



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of

Komatsu et al.

Serial No.: 10/797,800

Filed: March 10, 2004

For: COMPILATION AND RUNTIME INFORMATION GENERATION AND
OPTIMIZATION

Date: December 6, 2004

Group Art Unit: 2122

Examiner: Not yet assigned

Docket No.: JP920030038US1

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUBMISSION OF PRIORITY DOCUMENT

Sir:

Enclosed herewith is a certified copy of Japanese Application No. 2003-066010
filed March 12, 2003, in support of applicant's claim to priority under 35 U.S.C. 119.

Respectfully submitted,

By

Louis P. Herzberg

Reg. No. 41,500

Phone No. (914) 945-2885

IBM Corporation
Intellectual Property Law Dept.
P.O. Box 218
Yorktown Heights, NY 10598

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2003年 3月12日

出 願 番 号

Application Number:

特願2003-066010

ST.10/C]:

[JP2003-066010]

出 願 人

Applicant(s):

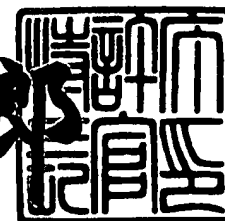
インターナショナル・ビジネス・マシーンズ・コーポレーション

CERTIFIED COPY OF
PRIORITY DOCUMENT

2003年 5月20日

特 許 庁 長 官
Commissioner,
Japan Patent Office

太田 信一郎



出証番号 出証特2003-3037036

【書類名】 特許願

【整理番号】 JP9030038

【提出日】 平成15年 3月12日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 9/45

【発明者】

 【住所又は居所】 神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ピー・エム株式会社 東京基礎研究所内

 【氏名】 安江 俊明

【発明者】

 【住所又は居所】 神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ピー・エム株式会社 東京基礎研究所内

 【氏名】 菅沼 俊夫

【発明者】

 【住所又は居所】 神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ピー・エム株式会社 東京基礎研究所内

 【氏名】 小松 秀昭

【特許出願人】

 【識別番号】 390009531

 【氏名又は名称】 インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】

 【識別番号】 100086243

 【弁理士】

 【氏名又は名称】 坂口 博

【代理人】

 【識別番号】 100091568

 【弁理士】

 【氏名又は名称】 市位 嘉宏

【代理人】

【識別番号】 100108501

【弁理士】

【氏名又は名称】 上野 剛史

【復代理人】

【識別番号】 100104156

【弁理士】

【氏名又は名称】 龍華 明裕

【手数料の表示】

【予納台帳番号】 053394

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9706050

【包括委任状番号】 9704733

【包括委任状番号】 0207860

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 コンパイラ装置、コンパイラプログラム、記録媒体、コンパイル方法、実行時情報生成装置、及び実行時情報生成プログラム

【特許請求の範囲】

【請求項 1】 最適化対象のプログラムにおける各処理が実行される頻度を収集し、収集した当該頻度に基づいて前記プログラムを最適化するコンパイラ装置であって、

前記プログラムのうち、繰り返し実行されるループ処理を検出するループ処理検出部と、

前記プログラムにおいて、前記ループ処理が実行されるループ処理頻度を収集するループ処理頻度収集部と、

前記ループ処理が実行される回数に対する、前記ループ処理に含まれる複数のループ内処理のそれぞれが実行される頻度であるループ内処理頻度を収集するループ内処理頻度収集部と、

前記ループ処理頻度及び前記ループ内処理頻度に基づき、前記プログラムが実行された場合における、前記複数のループ内処理のそれぞれが実行される頻度を示すループ内実行情報を生成するループ内実行情報生成部と、

前記ループ内実行情報生成部により生成された前記ループ内実行情報に基づき、前記プログラムを最適化する最適化部と
を備えるコンパイラ装置。

【請求項 2】 前記ループ内処理頻度収集部は、前記ループ処理頻度が予め定められた頻度より高い場合に、前記ループ内処理頻度を収集する請求項 1 記載のコンパイラ装置。

【請求項 3】 前記ループ内実行情報生成部は、前記ループ処理頻度と前記ループ内処理頻度を乗じることにより、前記ループ内実行情報を生成する請求項 1 記載のコンパイラ装置。

【請求項 4】 前記ループ処理は、更に内側のループ処理である内側ループ処理を含む外側ループ処理であり、

前記ループ処理検出部は、前記内側ループ処理を更に検出し、

前記ループ処理頻度収集部は、前記プログラムにおいて、前記内側ループ処理が実行されるループ処理頻度を前記ループ内実行情報に基づいて更に収集し、

前記ループ内処理頻度収集部は、前記内側ループ処理のループ内処理頻度を収集し、

前記ループ内実行情報生成部は、前記内側ループ処理におけるループ内処理頻度に、前記内側ループ処理のループ処理頻度を乗じることにより、前記内側ループ処理のループ内実行情報を生成する請求項 1 記載のコンパイラ装置。

【請求項 5】 前記ループ処理頻度収集部は、前記ループ処理が実行される回数を測定するカウンタを、前記プログラムが予め定められた回数実行された時点で停止して、当該カウンタにより測定された回数を前記ループ処理頻度として収集し、

前記ループ内処理頻度収集部は、前記複数のループ内処理のそれぞれが実行される回数を測定するカウンタを、前記複数のループ内処理の測定値の合計が予め定められた回数となった時点で停止する請求項 1 記載のコンパイラ装置。

【請求項 6】 前記プログラムにおける、複数の命令列のそれぞれをノードとして生成し、前記複数の命令列間の実行順序を前記ノード間の有向エッジとして生成した制御フローグラフを生成する制御フローグラフ生成部と、

前記制御フローグラフにおいて、前記ループ処理を形成するノードの集合に代えて、前記ループ処理全体を示す単一のループノードを生成した外郭構造グラフと、前記ループ処理を形成するノードの集合の制御フローグラフであるループ内構造グラフとを生成する構造グラフ生成部と、

前記外郭構造グラフ及び前記ループ内構造グラフのそれぞれにおいて、当該構造グラフにおける各実行パスが実行される回数をカウントするべく、前記プログラムにカウンタを挿入するカウンタ挿入部とを更に備え、

前記ループ処理頻度収集部は、前記プログラムが実行される回数に対する、前記ループノードが実行される回数を、前記ループ処理頻度として生成し、

前記ループ内処理頻度収集部は、前記ループ処理が実行される回数に対する、前記ループ内構造グラフにおける各実行パスが実行された回数を、前記ループ内

処理頻度として収集する請求項 1 記載のコンパイラ装置。

【請求項 7】 前記ループ処理頻度収集部は、前記プログラムが予め定められた回数実行された場合において、前記ループノードを含む実行パスが実行される回数をカウントするべく挿入されたカウンタの測定値を、ループ処理頻度として収集し、

前記ループ内処理頻度収集部は、前記複数のループ内処理の合計測定値が予め定められた回数となった場合において、前記ループ内構造グラフにおける各実行パスが実行される回数をカウントするべく挿入されたカウンタの測定値に基づき、ループ内処理頻度を収集する請求項 6 記載のコンパイラ装置。

【請求項 8】 前記カウンタ挿入部は、前記外郭構造グラフの各実行パスが実行される回数を測定するべくカウンタを挿入する前記プログラム中の挿入位置が、前記ループ内構造グラフの各実行パスが実行される回数を測定するべくカウンタを挿入する前記プログラム中の位置と同一であり、かつ前記外郭構造グラフ及び前記ループ内構造グラフのうち多くとも一方におけるカウンタが起動される場合に、当該挿入位置において、前記外郭構造グラフ及び前記ループ内構造グラフの双方における実行パスが実行される回数を測定するカウンタを挿入する請求項 6 記載のコンパイラ装置。

【請求項 9】 前記カウンタ挿入部は、前記外郭構造グラフの各実行パスが実行される回数を測定するべくカウンタを挿入する前記プログラム中の挿入位置が、前記ループ内構造グラフの各実行パスが実行される回数を測定するべくカウンタを挿入する前記プログラム中の位置と同一であり、かつ前記外郭構造グラフ及び前記ループ内構造グラフのうち多くとも一方におけるカウンタが起動される場合に、前記外郭構造グラフ及び前記ループ内構造グラフのそれぞれにおいて各実行パスの実行の回数を測定する複数の測定処理を生成し、

前記ループ内処理頻度収集部は、処理を他の部分に移すジャンプ命令を前記挿入位置に挿入し、当該ジャンプ命令のジャンプ先を、前記複数の測定処理の何れかに設定することにより、前記外郭構造グラフ及び前記ループ内構造グラフの双方における実行パスが実行される回数を測定する請求項 6 記載のコンパイラ装置。

【請求項 1 0】 前記ループ処理は、更に内側のループ処理である内側ループ処理を含む外側ループ処理であり、

前記ループ処理検出部は、前記内側ループ処理を更に検出し、

前記構造グラフ生成部は、前記外側ループ処理の制御フローグラフにおいて、前記内側ループ処理を形成するノードの集合に代えて、単一の内側ループノードを生成したグラフを外側ループ内構造グラフとして生成し、前記内側ループ処理を構成するノードの集合の制御フローグラフである内側ループ内構造グラフを生成し、

前記カウンタ挿入部は、前記内側ループ内構造グラフの各実行パスが実行される回数を測定するカウンタを更に挿入し、

前記ループ処理頻度収集部は、前記プログラムにおいて、前記内側ループ処理が実行されるループ処理頻度を前記ループ内実行情報に基づいて更に収集し、

前記ループ内処理頻度収集部は、前記内側ループ処理が実行される回数に対する、前記内側ループ内構造グラフにおける各実行パスが実行される頻度を、前記内側ループ処理におけるループ内処理頻度として収集し、

前記ループ内実行情報生成部は、前記内側ループ処理におけるループ内実行情報を、前記内側ループ処理におけるループ内処理頻度に、前記内側ループ処理のループ処理頻度を乗じることにより更に生成する請求項 6 記載のコンパイラ装置。

【請求項 1 1】 前記カウンタ挿入部は、前記外側ループ内構造グラフの各実行パスが実行される回数を測定するべくカウンタを挿入する前記プログラム中の挿入位置が、前記内側ループ内構造グラフの各実行パスが実行される回数を測定するべくカウンタを挿入する前記プログラム中の位置と同一であり、かつ前記外側ループ内構造グラフ及び前記内側ループ内構造グラフのうち多くとも一方におけるカウンタが起動される場合に、当該挿入位置において、前記外側ループ内構造グラフ及び前記内側ループ内構造グラフの双方における実行パスが実行される回数を測定するカウンタを挿入する請求項 1 0 記載のコンパイラ装置。

【請求項 1 2】 前記カウンタ挿入部は、前記外側ループ内構造グラフの各実行パスが実行される回数を測定するべくカウンタを挿入する前記プログラム中

の挿入位置が、前記内側ループ内構造グラフの各実行パスが実行される回数を測定するべくカウンタを挿入する前記プログラム中の位置と同一であり、かつ前記外側ループ内構造グラフ及び前記内側ループ内構造グラフのうち多くとも一方におけるカウンタが起動される場合に、前記外側ループ内構造グラフ及び前記内側ループ内構造グラフのそれぞれにおいて各実行パスの実行の回数を測定する複数の測定処理を生成し、

前記ループ内処理頻度収集部は、処理を他の部分に移すジャンプ命令を前記挿入位置に挿入し、当該ジャンプ命令のジャンプ先を、前記複数の測定処理の何れかに設定することにより、前記外側ループ内構造グラフ及び前記内側ループ内構造グラフの双方における実行パスが実行される回数を測定する請求項10記載のコンパイラ装置。

【請求項13】 最適化対象のプログラムにおける各処理が実行される頻度を収集し、収集した当該頻度に基づいて前記プログラムを最適化するコンパイラ装置としてコンピュータを機能させるコンパイラプログラムであって、

前記コンピュータを、

前記プログラムのうち、繰り返し実行されるループ処理を検出するループ処理検出部と、

前記プログラムにおいて、前記ループ処理が実行されるループ処理頻度を収集するループ処理頻度収集部と、

前記ループ処理が実行される回数に対する、前記ループ処理に含まれる複数のループ内処理のそれぞれが実行される頻度であるループ内処理頻度を収集するループ内処理頻度収集部と、

前記ループ処理頻度及び前記ループ内処理頻度に基づき、前記プログラムが実行された場合における、前記複数のループ内処理のそれぞれが実行される頻度を示すループ内実行情報を生成するループ内実行情報生成部と、

前記ループ内実行情報生成部により生成された前記ループ内実行情報に基づき、前記プログラムを最適化する最適化部として機能させるコンパイラプログラム。

【請求項14】 請求項13に記載のコンパイラプログラムを記録した記録

媒体。

【請求項 1 5】 最適化対象のプログラムにおける各処理が実行される頻度を収集し、収集した当該頻度に基づいて前記プログラムを最適化するコンパイル方法であって、

前記プログラムのうち、繰り返し実行されるループ処理を検出するループ処理検出段階と、

前記プログラムにおいて、前記ループ処理が実行されるループ処理頻度を収集するループ処理頻度収集段階と、

前記ループ処理が実行される回数に対する、前記ループ処理に含まれる複数のループ内処理のそれぞれが実行される頻度であるループ内処理頻度を収集するループ内処理頻度収集段階と、

前記ループ処理頻度及び前記ループ内処理頻度に基づき、前記プログラムが実行された場合における、前記複数のループ内処理のそれぞれが実行される頻度を示すループ内実行情報を生成するループ内実行情報生成段階と、

前記ループ内実行情報生成部により生成された前記ループ内実行情報に基づき、前記プログラムを最適化する最適化段階とを備えるコンパイル方法。

【請求項 1 6】 最適化対象のプログラムにおける各処理が実行される頻度を収集する実行時情報生成装置であって、

前記プログラムのうち、繰り返し実行されるループ処理を検出するループ処理検出部と、

前記プログラムにおいて、前記ループ処理が実行されるループ処理頻度を収集するループ処理頻度収集部と、

前記ループ処理が実行される回数に対する、前記ループ処理に含まれる複数のループ内処理のそれぞれが実行される頻度であるループ内処理頻度を収集するループ内処理頻度収集部と、

前記ループ処理頻度及び前記ループ内処理頻度に基づき、前記プログラムが実行された場合における、前記複数のループ内処理のそれぞれが実行される頻度を示すループ内実行情報を生成するループ内実行情報生成部と

を備え、前記ループ内実行情報生成部により生成された前記ループ内実行情報に基づき、前記プログラムを最適化させる実行時情報生成装置。

【請求項 1 7】 最適化対象のプログラムにおける各処理が実行される頻度を収集する実行時情報生成装置としてコンピュータを機能させる実行時情報生成プログラムであって、

前記コンピュータを、

前記プログラムのうち、繰り返し実行されるループ処理を検出するループ処理検出部と、

前記プログラムにおいて、前記ループ処理が実行されるループ処理頻度を収集するループ処理頻度収集部と、

前記ループ処理が実行される回数に対する、前記ループ処理に含まれる複数のループ内処理のそれぞれが実行される頻度であるループ内処理頻度を収集するループ内処理頻度収集部と、

前記ループ処理頻度及び前記ループ内処理頻度に基づき、前記プログラムが実行された場合における、前記複数のループ内処理のそれぞれが実行される頻度を示すループ内実行情報を生成するループ内実行情報生成部と

して機能させ、前記ループ内実行情報生成部により生成された前記ループ内実行情報に基づき、前記プログラムを最適化させる実行時情報生成プログラム。

【請求項 1 8】 請求項 1 7 に記載の実行時情報生成プログラムを記録した記録媒体。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、コンパイラ装置、コンパイラプログラム、記録媒体、コンパイル方法、実行時情報生成装置、及び実行時情報生成プログラムに関する。特に本発明は、プログラムを実行させた場合に得られる実行情報を用いて最適化を行うコンパイラ装置、コンパイラプログラム、記録媒体、コンパイル方法、実行時情報生成装置、及び実行時情報生成プログラムに関する。

【0 0 0 2】

【従来の技術】

従来、プログラムにおける複数の実行パスのそれぞれが実行される回数を収集する技術が用いられている。例えば、非特許文献1に記載の技術によると、コンパイラは、複数の実行パスが実行される回数をカウントするべく、カウンタを適切な位置に挿入することができる。

【0003】

【非特許文献1】

Ball, T及びLarus, J. R. 著”Efficient Path Profiling”第29回マイクロアーキテクチャ国際会議(MICRO-29)会議録、46ページ-57ページ、平成8年12月、邦題「効率的パスプロファイル」

【0004】

【発明が解決しようとする課題】

しかしながら、上記技術は、各実行パスが実行される回数を適切に収集できるものの、収集の処理に長い時間を要する。

そこで本発明は、上記の課題を解決することのできるコンパイラ装置、コンパイラプログラム、記録媒体、コンパイル方法、実行時情報生成装置、及び実行時情報生成プログラムを提供することを目的とする。この目的は特許請求の範囲における独立項に記載の特徴の組み合わせにより達成される。また従属項は本発明の更なる有利な具体例を規定する。

【0005】

【課題を解決するための手段】

即ち、本発明の第1の形態によると、最適化対象のプログラムにおける各処理が実行される頻度を収集し、収集した当該頻度に基づいてプログラムを最適化するコンパイラ装置は、プログラムのうち、繰り返し実行されるループ処理を検出するループ処理検出部と、プログラムにおいて、ループ処理が実行されるループ処理頻度を収集するループ処理頻度収集部と、ループ処理が実行される回数に対する、ループ処理に含まれる複数のループ内処理のそれぞれが実行される頻度であるループ内処理頻度を収集するループ内処理頻度収集部と、ループ処理頻度及

びループ内処理頻度に基づき、プログラムが実行された場合における、複数のループ内処理のそれぞれが実行される頻度を示すループ内実行情報を生成するループ内実行情報生成部と、ループ内実行情報生成部により生成されたループ内実行情報に基づき、プログラムを最適化する最適化部とを備えるコンパイラ装置、コンパイル方法、コンパイラプログラム、実行時情報生成装置、実行時情報生成プログラム、及び記録媒体を提供する。

なお上記の発明の概要は、本発明の必要な特徴の全てを列挙したものではなく、これらの特徴群のサブコンビネーションも又発明となりうる。

【 0 0 0 6 】

【発明の実施の形態】

以下、発明の実施の形態を通じて本発明を説明するが、以下の実施形態は特許請求の範囲にかかる発明を限定するものではなく、又実施形態の中で説明されている特徴の組み合わせの全てが発明の解決手段に必須であるとは限らない。

図 1 は、コンパイラ装置 1 0 の機能ブロック図を示す。コンパイラ装置 1 0 は、最適化対象のプログラムにおける各処理が実行される頻度を収集する実行時情報生成装置 2 0 と、実行時情報生成装置 2 0 により収集された当該頻度に基づいてプログラムを最適化する最適化部 3 0 とを備える。実行時情報生成装置 2 0 は、制御フローグラフ生成部 1 0 0 と、ループ検出部 1 1 0 と、構造グラフ生成部 1 2 0 と、カウンタ挿入部 1 3 0 と、ループ処理頻度収集部 1 4 0 と、ループ内処理頻度収集部 1 5 0 と、ループ内実行情報生成部 1 6 0 とを備え、ループ内実行情報生成部 1 6 0 により生成されたループ内実行情報に基づき、最適化部 3 0 によりプログラムを最適化させる。

【 0 0 0 7 】

制御フローグラフ生成部 1 0 0 は、コンパイル対象のプログラムを受け取ると、当該プログラムにおける、複数の命令列のそれぞれをノードとして生成し、当該複数の命令列間の実行順序をノード間の有向エッジとして生成した制御フローグラフを生成する。そして、制御フローグラフ生成部 1 0 0 は、制御フローグラフをプログラムと共にループ検出部 1 1 0 に送る。

【 0 0 0 8 】

なお、コンパイル対象のプログラムは、最適化を効率良く行うべくソースプログラムから生成された中間表現であり、例えば、Java（登録商標）言語のバイトコードである。これに代えて、プログラムは、RTL（Register Transfer Language）であってもよいし、四つ組表現であってもよい。

また、命令列とは、連続して実行される命令の集合である。一例としては、命令列は、最初に実行される命令及び最後に実行される命令以外の命令が、分岐命令の分岐元及び分岐命令の分岐先の何れでもない命令の集合である基本ブロック（Basic Block）である。他の例としては、命令列は、最初に実行される命令及び最後に実行される命令以外の命令が、分岐命令の分岐先とならない命令の集合であるスーパーブロック（Super Block）であってよい。

【0009】

ループ検出部110は、制御フローグラフ及びプログラムを制御フローグラフ生成部100から受け取ると、プログラムのうち、繰り返し実行されるループ処理を検出する。ループ検出部110は、検出したループ処理が、更に内側のループ処理である内側ループ処理を含む場合に、当該内側ループ処理を更に検出する。そして、ループ検出部110は、検出したループ処理の情報を、制御フローグラフ及びプログラムと共に、構造グラフ生成部120に送る。なお、ループ処理とは、制御フローグラフにおいて、互いに到達可能なノードの集合である強連結成分に対応する命令の集合である。

【0010】

構造グラフ生成部120は、制御フローグラフにおいて、外側ループ処理を形成するノードの集合に代えて、外側ループ処理全体を示す単一のノードである外側ループノードを生成した外郭構造グラフを生成する。また、構造グラフ生成部120は、外側ループ処理の制御フローグラフにおいて、内側ループ処理を形成するノードの集合に代えて、内側ループ処理全体を示す単一のノードである内側ループノードを生成した外側ループ内構造グラフを生成する。更に、構造グラフ生成部120は、内側ループ処理の制御フローグラフである内側ループ内構造グラフを生成する。そして、構造グラフ生成部120は、外郭構造グラフ、外側ル

ープ内構造グラフ、内側ループ内構造グラフ、及びプログラムを、カウンタ挿入部 1 3 0 に送る。

【 0 0 1 1 】

カウンタ挿入部 1 3 0 は、外郭構造グラフ、外側ループ内構造グラフ、及び内側ループ内構造グラフのそれぞれにおいて、当該構造グラフにおける各実行パスが実行される回数をカウントするべく、プログラムにカウンタを挿入する。そして、カウンタ挿入部 1 3 0 は、カウンタを挿入したプログラムを、外郭構造グラフ、外側ループ内構造グラフ、及び内側ループ内構造グラフと共にループ処理頻度収集部 1 4 0 に送る。

【 0 0 1 2 】

ループ処理頻度収集部 1 4 0 は、外郭構造グラフ、外側ループ内構造グラフ、及び内側ループ内構造グラフをカウンタ挿入部 1 3 0 から受け取る。そして、ループ処理頻度収集部 1 4 0 は、カウンタを挿入したプログラムをカウンタ挿入部 1 3 0 から受け取った場合に、外郭構造グラフの各実行パスが実行される回数をカウントするべく挿入されたカウンタを起動し、受け取ったプログラムを実行する。その後、ループ処理頻度収集部 1 4 0 は、起動したカウンタを、プログラムが予め定められた回数実行された時点で停止する。そして、ループ処理頻度収集部 1 4 0 は、停止した時点でカウンタにより測定された、外側ループ処理が実行された回数を、外側ループ処理が実行される外側ループ処理頻度として収集し、収集結果をプログラムと共にループ内処理頻度収集部 1 5 0 に送る。また、ループ処理頻度収集部 1 4 0 は、プログラムが実行された回数に対する、外郭構造グラフの各実行パスが実行された頻度である外郭構造グラフ頻度情報を、プログラムと共に、最適化部 3 0 に送る。

【 0 0 1 3 】

好ましくは、ループ処理頻度収集部 1 4 0 は、タイマー・サンプリング・プロファイラ等の、プログラムの実行頻度を測定する装置を用いて、より頻繁に実行されるプログラム片を検出し、当該プログラム片の外郭構造グラフについてのみ、カウンタを起動する。ここで、プログラム片とは、例えば、メソッド (method)、関数 (function)、又は手続 (procedure) である

。この場合、より頻繁に実行されるプログラム片を優先して最適化することができるので、プログラムの処理速度を向上できると共に、コンパイラ装置 1 0 を高速に動作させることができる。

【 0 0 1 4 】

一方、ループ処理頻度収集部 1 4 0 は、外側ループ処理のループ内実行情報をループ内実行情報生成部 1 6 0 から受け取った場合に、当該ループ内実行情報に基づき、内側ループ処理のループ処理頻度を収集し、収集結果をループ内処理頻度収集部 1 5 0 に送る。

【 0 0 1 5 】

ループ内処理頻度収集部 1 5 0 は、外側ループ処理頻度の収集結果をループ処理頻度収集部 1 4 0 から受け取ると、外側ループ処理の処理頻度が、予め定められた基準頻度より高いか否かを判断する。外側ループ処理の処理頻度が基準頻度より高い場合に、ループ内処理頻度収集部 1 5 0 は、外側ループ内構造グラフの各実行パスが実行される回数をカウントするべく挿入されたカウンタを起動することにより、外側ループ処理における複数のループ内処理のそれぞれが実行される回数を測定する。その後、ループ内処理頻度収集部 1 5 0 は、起動としておいたカウンタを、複数のループ内処理の測定値の合計が予め定められた回数となった時点で停止する。そして、ループ内処理頻度収集部 1 5 0 は、停止したカウンタの測定値に基づいて、外側ループ内処理が実行される回数に対する、各外側ループ内処理が実行された回数を、外側ループ内処理頻度として収集し、収集結果をループ内実行情報生成部 1 6 0 に送る。

【 0 0 1 6 】

ループ内処理頻度収集部 1 5 0 が、内側ループ処理頻度の収集結果をループ処理頻度収集部 1 4 0 から受け取った場合の処理は、ループ内処理頻度収集部 1 5 0 が、外側ループ処理に対して行う処理と略同一であるので説明を省略する。

【 0 0 1 7 】

ここで、例えば、ループ内処理とは、構造グラフの実行パス上にある命令の集合である。これに代えて、ループ内処理とは、構造グラフの各ノードが示す命令列であってもよいし、構造グラフ中の分岐命令であってもよい。即ち、ループ内

処理頻度収集部 1 5 0 は、プログラム中の命令から、予め設定した方法で複数の命令グループを生成し、各命令グループが実行される回数を各ループ内処理が実行される回数として測定する。

【 0 0 1 8 】

ループ内実行情報生成部 1 6 0 は、外側ループ処理頻度及び外側ループ内処理頻度を受け取った場合に、プログラムが実行された場合における、複数の外側ループ内処理のそれぞれが実行される頻度を示すループ内実行情報を生成し、ループ処理頻度収集部 1 4 0 及び最適化部 3 0 に送る。また、ループ内実行情報生成部 1 6 0 は、内側ループ処理頻度及び内側ループ内処理頻度を受け取った場合に、プログラムが実行された場合における、複数の内側ループ内処理のそれぞれが実行される頻度を示すループ内実行情報を生成し、最適化部 3 0 に送る。

【 0 0 1 9 】

なお、ループ内実行情報生成部 1 6 0 は、内側ループ処理の更に内側にループ処理がある場合には、内側ループ処理のループ内実行情報を、ループ処理頻度収集部 1 4 0 に更に送ってもよい。この場合、ループ処理頻度収集部 1 4 0 及びループ内処理頻度収集部 1 5 0 は、内側ループ処理の更に内側にあるループ処理について略同一の動作を繰り返す。

【 0 0 2 0 】

最適化部 3 0 は、ループ内実行情報及び外郭構造グラフ頻度情報に基づき、ループ処理頻度収集部 1 4 0 から受け取ったプログラムを最適化する。なお、これに代えて、最適化部 3 0 は、カウンタ挿入部 1 3 0 によりカウンタが挿入される以前のプログラムを最適化してもよい。そして、最適化部 3 0 は、最適化した後のプログラムを、コンパイル結果のプログラムとして出力する。

【 0 0 2 1 】

図 2 は、コンパイラ装置 1 0 のフローチャートを示す。制御フローグラフ生成部 1 0 0 は、コンパイル対象のプログラムを受け取ると、当該プログラムの制御フローグラフを生成する (S 2 0 0) 。そして、ループ検出部 1 1 0 は、プログラムのうち、繰り返し実行されるループ処理を検出する (S 2 1 0) 。ループ検出部 1 1 0 は、検出したループ処理が、更に内側のループ処理である内側ループ

処理を含む場合に、当該内側ループ処理を更に検出する。

【 0 0 2 2 】

構造グラフ生成部 1 2 0 は、制御フローグラフにおいて、外側ループ処理を形成するノードの集合に代えて、外側ループ処理全体を示す単一のノードである外側ループノードを生成した外郭構造グラフを生成する（S 2 2 0）。また、構造グラフ生成部 1 2 0 は、外側ループ処理の制御フローグラフにおいて、内側ループ処理を形成するノードの集合に代えて、内側ループ処理全体を示す単一のノードである内側ループノードを生成した外側ループ内構造グラフを生成する。更に、構造グラフ生成部 1 2 0 は、内側ループ処理の制御フローグラフである内側ループ内構造グラフを生成する。

【 0 0 2 3 】

カウンタ挿入部 1 3 0 は、外郭構造グラフ、外側ループ内構造グラフ、及び内側ループ内構造グラフのそれぞれにおいて、当該構造グラフにおける各実行パスが実行される回数をカウントするべく、プログラムにカウンタを挿入する（S 2 3 0）。そして、ループ処理頻度収集部 1 4 0 は、プログラムを実行することにより、ループ処理の処理頻度、例えば、外側ループ処理頻度を生成する（S 2 4 0）。また、ループ処理頻度収集部 1 4 0 は、外側ループ処理のループ内処理頻度に基づいて内側ループ処理頻度を生成する。ループ内処理頻度収集部 1 5 0 は、ループ処理の処理頻度が、予め定められた基準頻度より高い場合に（S 2 5 0 : Y E S）、ループ処理が実行される回数に対する、ループ処理における複数のループ内処理のそれぞれが実行されるループ内処理頻度を収集する（S 2 6 0）。ループ内実行情報生成部 1 6 0 は、ループ処理頻度及びループ内処理頻度に基づいて、ループ内実行情報を生成する（S 2 7 0）。ループ内実行情報生成部 1 6 0 は、ループ処理が更に内側にループ処理を含む場合に（S 2 8 0 : Y E S）、内側のループ処理の処理頻度を収集するべく、S 2 4 0 に処理を移す。

【 0 0 2 4 】

ループ処理の処理頻度が、予め定められた基準頻度より低い場合（S 2 5 0 : N O）、又は、ループ処理が更に内側にループ処理を含まない場合に（S 2 8 0 : N O）、最適化部 3 0 は、ループ内実行情報に基づき、プログラムを最適化し

、コンパイル結果のプログラムとして出力する（S 2 9 0）。

【 0 0 2 5 】

なお、ループ内処理頻度収集部 1 5 0 が、ループ内処理頻度を収集するタイミングは、本フローチャートにおけるタイミングに限定されない。例えば、ループ処理頻度収集部 1 4 0 が、外郭構造グラフの各実行パスが実行される回数の収集を開始する時点で、ループ内処理頻度収集部 1 5 0 は、外側ループ内構造グラフ及び内側ループ内構造グラフのそれぞれについても、ループ内処理頻度の収集を開始してもよい。また、他の例としては、外側ループ内構造グラフ及び内側ループ内構造グラフ間に、ループ内処理頻度を収集する順序を予め定めておき、ループ内処理頻度収集部 1 5 0 は、当該予め定められた順序に応じて、ループ内処理頻度を収集してもよい。

【 0 0 2 6 】

更に他の例としては、プログラム内に複数の外側ループ処理が存在する場合には、ループ内処理頻度収集部 1 5 0 は、ループ処理の階層の深さ毎に、ループ内処理頻度の収集を開始してもよい。例えば、ループ内処理頻度収集部 1 5 0 は、一の外側ループ処理におけるループ内処理頻度の収集を開始する場合には、他の外側ループ処理におけるループ内処理頻度の収集を開始し、一の内側ループ処理におけるループ内処理頻度の収集を開始する場合には、他の内側ループ処理におけるループ内処理頻度の収集を開始してもよい。

【 0 0 2 7 】

図 3 は、最適化対象のプログラムの一例を示す。この最適化対象のプログラムは、1 行目から 1 4 行目の文が示すメソッド「m」を有する。メソッド「m」は、4 行目から 1 2 行目までの外側ループ処理を有する。そして、外側ループ処理は、6 行目から 1 0 行目までの内側ループ処理を有する。9 行目の文は、8 行目に示した条件が成立した場合に、メソッド「m」の処理を終了する処理を示す。

【 0 0 2 8 】

また、3 行目の文、5 行目の文、6 行目の文、7 から 8 行目の文、9 行目の文、1 1 から 1 2 行目の文、及び 1 3 行目の文は、それぞれ、第 1 から第 7 基本ブロックを構成する。

【0029】

図4は、制御フローグラフの一例を示す。制御フローグラフ生成部100は、図3に示したプログラムを受け取ると、本図で示す制御フローグラフを生成する。本図において、丸印は、プログラムの命令列を示すノードであり、矢印は、命令列間の実行順序を示す有向エッジである。また、本図における命令列は基本ブロックであり、ノード中に記載の番号は、図3で示した基本ブロックを識別するノード番号である。一例としては、第3ノードから第4ノード及び第6ノードに向かう有向エッジは、第3基本ブロックの後に、第4基本ブロック又は第6基本ブロックが実行される旨を示している。

【0030】

図5(a)は、構造グラフを生成する対象となる制御フローグラフの一例を示す。本図に示した制御フローグラフによると、ループ前処理が行われた後に、ヘッダノード500及びラッチノード510が順次実行される。続いて、ラッチノード510の処理結果に応じて、ヘッダノード500又はループ後処理が実行される。即ち、ヘッダノード500及びラッチノード510は、ループ処理を形成している。

【0031】

図5(b)は、制御フローグラフの実行パスを示す。図5(a)に示した制御フローグラフは、ループ前処理からヘッダノード500及びラッチノード510を順次実行し繰り返し実行することなくループ後処理に移る実行パス520と、ループ前処理からヘッダノード500及びラッチノード510を順次実行し繰り返し処理に移る実行パス530と、ラッチノード510からヘッダノード500に至る実行パスが更にラッチノード510を処理する実行パス540と、ヘッダノード500及びラッチノード510を順次実行した後にループ後処理に移る実行パス550とを有する。

【0032】

図5(c)は、制御フローグラフから生成した構造グラフの実行パスを示す。構造グラフ生成部120は、外郭構造グラフと、ループ内構造グラフとを生成する。ループ内構造グラフは、実行パス530と、実行パス540と、実行パス5

50と、ヘッダノード500からラッチノード510に至る制御フロー560とを有する。外郭構造グラフは、ループ前処理からループ処理を経てループ後処理に至る実行パス520を有する。このように、構造グラフ生成部120は、外郭構造グラフとして、制御フローグラフにおいて、ループ処理を形成するノードの集合に代えて、ループ処理全体を単一のループノードとして生成したグラフを生成する。即ち、本図における外郭構造グラフの実行パスは、実行パス520である。また、構造グラフ生成部120は、ループ内構造グラフとして、ループ処理を形成するノードの集合の制御フローグラフを生成する。即ち、本図におけるループ内構造グラフの実行パスは、実行パス530、実行パス540、実行パス550、及び制御フロー560である。

【0033】

より具体的には、構造グラフ生成部120は、ループ内構造グラフを生成するべく、以下の処理を行う。構造グラフ生成部120は、ループ処理の外からループ処理が開始されるエントリーノードであるヘッダノード500と、ループ処理からループ処理の外に処理を移すエグジットノードであるラッチノード510とを、ループ内構造グラフとして生成する。そして、構造グラフ生成部120は、制御フローグラフにおいてヘッダノード500からラッチノード510に至る全てのエッジ及びノードを、ループ内構造グラフに含める。そして、構造グラフ生成部120は、ループ前処理に代えて、ループ内構造グラフの開始点を示すダミーノードから、ヘッダノード500にエッジを生成する。また、構造グラフ生成部120は、ラッチノード510から、ループ後処理に代えて、ループ内構造グラフの終了点を示すダミーノードへのエッジを生成する。

【0034】

図6(a)は、図4に示した制御フローグラフから生成される外郭構造グラフの一例を示す。本図における太線の丸印は、ループ処理に代えて生成されたループノードを示す。例えば、構造グラフ生成部120は、以下の処理により外郭構造グラフを生成する。構造グラフ生成部120は、図4に示した制御フローグラフにおいて、外側ループ処理を形成するノードの集合、即ち、第2ノード、第3ノード、第4ノード、及び第6ノードに代えて、外側ループ処理全体を示す外側

ループノードである第2ノードを生成する。

【0035】

続いて、構造グラフ生成部120は、第2ノードから、第2ノードの後に実行される第5ノード及び第7ノードに向けて、点線の有向エッジを生成する。ここで、点線の有向エッジは、図4で示した制御フローグラフに実際には存在せず、制御フローグラフにおける複数の有向エッジの組合せにより実現される仮想的な実行経路を示している。例えば、第2ノードから第5ノードに向かう有向エッジは、外側ループの実行の後に第4ノードを介して第5ノードに至る実行経路を示している。なお、外郭構造グラフの開始点及び終了点を示す仮想的なノードを、それぞれ、E1及びX1と示した。

【0036】

図6(b)は、図4に示した制御フローグラフから生成される外側ループ内構造グラフの一例を示す。構造グラフ生成部120は、以下の処理により外側ループ内構造グラフを生成する。構造グラフ生成部120は、外側ループ処理の制御フローグラフにおいて、内側ループ処理を構成するノードの集合、即ち、第3ノード及び第4ノードに代えて、内側ループ処理全体を示す内側ループノードである第3ノードを生成する。

【0037】

そして、構造グラフ生成部120は、外側ループ内構造グラフの開始点E2から外側ループ処理の入り口である第2ノードへ至る点線の有向エッジと、第3ノードから外側ループ内構造グラフの終了点X2に至る点線の有向エッジと、第6ノードからX2に至る点線の有向エッジとを生成する。

【0038】

図6(c)は、図4に示した制御フローグラフから生成される内側ループ内構造グラフの一例を示す。構造グラフ生成部120は、以下の処理により内側ループ内構造グラフを生成する。構造グラフ生成部120は、内側ループ処理を構成するノードの集合の制御フローグラフを生成する。そして、構造グラフ生成部120は、内側ループ内構造グラフの開始点E3から内側ループ処理の入り口である第3ノードへ至る点線の有向エッジと、第3ノードから内側ループ内構造グラ

フの終了点X3に至る点線の有向エッジと、第4ノードからX3に至る点線の有向エッジとを生成する。

【0039】

なお、本図の例とは異なり、コンパイル対象のプログラムが、既約グラフ (irreducible graph) である場合には、構造グラフ生成部120は、各構造グラフの開始点を示すノードから、ループ処理の開始点となり得る複数のノードのそれぞれに対して、有向エッジを生成する。

【0040】

カウンタ挿入部130は、以上のように生成された構造グラフのそれぞれにおいて、各実行パスが実行される回数をカウントするべく、プログラムにカウンタを挿入する。カウンタ挿入部130により挿入されたカウンタの位置の一例を黒点で示した。例えば、外郭構造グラフにおいて、カウンタ挿入部130は、ノード4からノード5に至る有向エッジと、ノード6からノード7に至る有向エッジとに、カウンタを挿入する。また、外側ループ内構造グラフにおいて、カウンタ挿入部130は、ノード1からノード2に至る有向エッジと、ノード4からノード5に至る有向エッジと、ノード6からノード2に至る有向エッジと、ノード6からノード7に至る有向エッジとにカウンタを挿入する。また、内側ループ内構造グラフにおいて、カウンタ挿入部130は、ノード2からノード3に至る有向エッジと、ノード3からノード6に至る有向エッジと、ノード4からノード5に至る有向エッジとに、カウンタを挿入する。

【0041】

なお、カウンタを挿入する位置は、本図の例に限定されない。例えば、非特許文献1には、各実行パスが実行される回数を効率的に測定する方法が提案されており、当該方法を用いてカウンタを挿入する位置を決定してもよい。即ち、カウンタ挿入部130は、各構造グラフにおいて、各実行パスが実行される回数をカウントできる位置に、カウンタを挿入する。また、カウンタ挿入部130は、必要に応じて、カウンタを初期化する初期化処理を挿入してもよい。また、プログラムに複数のカウンタが挿入される場合には、カウンタ挿入部130は、当該複数のカウンタの中から、測定対象のカウンタを変更する処理をプログラムに更に

挿入してもよい。例えば、カウンタ挿入部 1 3 0 は、当該複数のカウンタを、それぞれが 1 つのカウンタである配列変数として生成した場合には、測定対象のカウンタを変更するべく、配列変数の添え字を変更する処理をプログラムに更に挿入してもよい。即ち、カウンタ挿入部 1 3 0 は、各実行パスが実行される回数をカウントするべく、カウンタを制御する処理を、プログラムに挿入する。

【 0 0 4 2 】

図 7 (a) は、プログラムに挿入されたカウンタが停止されている例を示す。カウンタ挿入部 1 3 0 は、構造グラフの各実行パスが実行される回数をカウントするべくカウンタを挿入するプログラム中の挿入位置に、NOP 命令 7 0 0 を挿入する。そして、カウンタ挿入部 1 3 0 は、実行の回数を測定する測定処理 7 1 0 を生成する。カウンタ挿入部 1 3 0 は、NOP 命令 7 0 0 の直後に実行される命令に処理を移すジャンプ命令を、測定処理 7 1 0 の最後に実行される箇所に生成する。

なお、より詳しくは、カウンタ挿入部 1 3 0 は、コンパイル対象のプログラムにおける基本ブロック内に、NOP 命令 7 0 0 又はジャンプ命令 7 2 0 を挿入する。しかしながら、測定対象である実行パスによっては、NOP 命令 7 0 0 又はジャンプ命令 7 2 0 を既存の基本ブロックに挿入できない場合がある。この場合、カウンタ挿入部 1 3 0 は、新たな基本ブロック、例えば、NOP 命令 7 0 0 又はジャンプ命令 7 2 0 等の、カウンタを実現する命令を挿入する基本ブロックを、生成してもよい。

【 0 0 4 3 】

図 7 (b) は、プログラムに挿入されたカウンタが起動されている例を示す。ループ内実行情報生成部 1 6 0 は、NOP 命令 7 0 0 に代えて、処理を測定処理 7 1 0 にジャンプさせるジャンプ命令 7 2 0 を生成する。これにより、ループ内実行情報生成部 1 6 0 は、ジャンプ命令 7 2 0 を含む実行パスが実行された回数を、測定処理 7 1 0 により測定させることができる。

【 0 0 4 4 】

図 7 (c) は、同一の挿入位置に複数のカウンタを生成する例を示す。本図を用いて、外郭構造グラフの各実行パスが実行される回数を測定するべくカウンタ

を挿入するプログラム中の挿入位置が、ループ内構造グラフの各実行パスが実行される回数を測定するべくカウンタを挿入するプログラム中の位置と同一であり、かつ外郭構造グラフ及びループ内構造グラフのうち多くとも一方におけるカウンタが起動される場合における、コンパイラ装置10の処理を説明する。

【0045】

カウンタ挿入部130は、外郭構造グラフ及びループ内構造グラフのそれぞれにおいて各実行パスが実行される回数を測定する複数の測定処理を生成する。例えば、カウンタ挿入部130は、外郭構造グラフの各実行パスが実行される回数を測定する測定処理710と、ループ内構造グラフの各実行パスが実行される回数を測定する測定処理730とを生成する。更に、カウンタ挿入部130は、カウンタの挿入位置（例えば、ジャンプ命令720の位置）の次に実行される命令にジャンプするジャンプ命令を、測定処理710及び測定処理730のそれぞれの最後に実行される位置に挿入する。

【0046】

ループ内実行情報生成部160は、処理を測定処理710にジャンプさせるジャンプ命令720を、カウンタの挿入位置に生成することにより、外郭構造グラフの各実行パスが実行される回数を測定させる。また、ループ内実行情報生成部160は、処理を測定処理730にジャンプさせるジャンプ命令720を、カウンタの挿入位置に生成することにより、ループ内構造グラフ各実行パスが実行される回数を測定させる。このように、カウンタ挿入部130は、ジャンプ命令のジャンプ先を、複数の測定処理の何れかに設定することにより、当該挿入位置において、外郭構造グラフ及びループ内構造グラフの双方における実行パスが実行される回数を測定することができる。

【0047】

また、外側ループ内構造グラフの各実行パスが実行される回数を測定するべくカウンタを挿入するプログラム中の挿入位置が、内側ループ内構造グラフの各実行パスが実行される回数を測定するべくカウンタを挿入するプログラム中の位置と同一であり、かつ外側ループ内構造グラフ及び内側ループ内構造グラフのうち多くとも一方におけるカウンタが起動される場合においても、以下に示すように

、コンパイラ装置 1 0 は、本図と略同一に動作する。

【 0 0 4 8 】

具体的には、カウンタ挿入部 1 3 0 は、外側ループ内構造グラフ及び内側ループ内構造グラフのそれぞれにおいて各実行パスが実行される回数を測定する複数の測定処理を生成する。例えば、カウンタ挿入部 1 3 0 は、外側ループ内構造グラフの各実行パスが実行される回数を測定する測定処理 7 1 0 と、内側ループ内構造グラフの各実行パスが実行される回数を測定する測定処理 7 3 0 とを生成する。更に、カウンタ挿入部 1 3 0 は、カウンタの挿入位置（例えば、ジャンプ命令 7 2 0 の位置）の次に実行される命令にジャンプするジャンプ命令を、測定処理 7 1 0 及び測定処理 7 3 0 のそれぞれの最後に実行される位置に挿入する。

【 0 0 4 9 】

そして、ループ内実行情報生成部 1 6 0 は、処理を測定処理 7 1 0 にジャンプさせるジャンプ命令 7 2 0 を、カウンタの挿入位置に生成することにより、外側ループ内構造グラフの各実行パスが実行される回数を測定させる。また、ループ内実行情報生成部 1 6 0 は、処理を測定処理 7 3 0 にジャンプさせるジャンプ命令 7 2 0 を、カウンタの挿入位置に生成することにより、内側ループ内構造グラフ各実行パスが実行される回数を測定させる。このように、カウンタ挿入部 1 3 0 は、ジャンプ命令のジャンプ先を、複数の測定処理の何れかに設定することにより、当該挿入位置において、外側ループ内構造グラフ及び内側ループ内構造グラフの双方における実行パスが実行される回数を測定することができる。

【 0 0 5 0 】

以上、本図で示したように、カウンタ挿入部 1 3 0 は、2 つの構造グラフのうち多くとも一方におけるカウンタが起動される場合、即ち、2 つの構造グラフの双方において同時にカウンタが使用されないことが保証される場合に、挿入位置において、2 つの構造グラフの双方における実行パスが実行される回数を測定することができる。3 つ以上の構造グラフについても同様に、コンパイラ装置 1 0 は、多くとも一の構造グラフにおけるカウンタのみが起動される場合には、カウンタを共用することができる。

なお、カウンタ挿入部 1 3 0 は、同時に使用される 2 つのカウンタについては

、当該2つのカウンタのそれぞれを挿入位置に生成する。例えば、外郭構造グラフにおいて各実行パスが実行される回数の収集が開始された時点で、外側ループ内構造グラフにおけるループ内処理頻度の収集を開始する場合には、カウンタ挿入部130は、外郭構造グラフ及び外側ループ内構造グラフのそれぞれにおけるカウンタのそれぞれを、挿入位置に生成する。このように、カウンタ挿入部130は、ループ内処理頻度を収集するタイミングに応じて、カウンタを挿入する方式を変更してもよい。

【0051】

また、測定処理の実行に排他制御が行われず、複数のスレッドが同時に測定処理を実行する場合には、カウンタの値が不正確となる場合がある。しかしながら、スレッドの数が測定値に比べて充分小さい場合には、測定値の誤差はわずかなので、コンパイラ装置10は、各実行パスが実行される回数を略正確に測定することができる。

【0052】

図8は、コンパイラ装置10が生成した実行情報の一例を示す。より詳細には、本図は、実行パスを識別する識別番号に、実行パスを構成するノードの順列と、カウンタにより測定された実行パスの実行回数である測定値と、測定値に基づき生成された実行情報（例えば、ループ内実行情報）と、実際に実行された実行回数とを対応付けて、構造グラフ毎に示す。

【0053】

ループ処理頻度収集部140は、プログラムを予め定められた100回実行した時点で、外郭構造グラフの実行パスが実行される回数を測定するカウンタを停止する。この時点で、ノードE1から、第1ノード、第2ノード、及び第7ノードを順次実行し、ノードX1に至る第1実行パスが実行された回数は、100回と測定される。即ち、第2実行パスは一度も実行されていない。この場合、ループ内実行情報生成部160は、プログラムが100回実行された場合における、第1パスが実行される頻度である実行情報として100.0を生成する。

【0054】

続いて、ループ内処理頻度収集部150は、外側ループ処理頻度、即ち、第2

ノードが実行される頻度が、予め定められた頻度より高い場合に、外側ループ内処理頻度を収集する。まず、ループ内処理頻度収集部 1 5 0 は、外側ループ内構造グラフにおける複数の実行パスのそれぞれが実行される回数を測定するカウンタを起動し、複数の実行パスにおける測定値の合計が予め定められた 1 0 0 回となった時点で停止する。そして、ループ内処理頻度収集部 1 5 0 は、外側ループ処理が実行される回数に対する、各実行パスが実行される頻度である外側ループ内処理頻度を収集する。

【 0 0 5 5 】

例えば、外側ループ処理が実行された回数は、E 2 ノードから外側ループに処理が移った回数であるので、第 3 から第 5 実行パスの合計値である 5 1 回となる。一方、例えば、第 8 実行パスが実行された回数は、4 8 回である。即ち、ループ内処理頻度収集部 1 5 0 は、外側ループ内処理頻度として、外側ループ処理が 5 1 回実行された場合に、第 8 実行パスが 4 8 回実行される旨を示す情報を収集する。

【 0 0 5 6 】

そして、ループ内実行情報生成部 1 6 0 は、外側ループ処理の処理頻度である 1 0 0 . 0 に、外側ループ内処理頻度、例えば、第 8 実行パスの処理頻度である $48 / 51$ を乗じることにより、外側ループ処理のループ内実行情報である 9 4 . 1 を生成する。ループ内実行情報生成部 1 6 0 は、第 3 実行パスから第 7 実行パスにおいても、第 8 実行パスにおける方法と略同一の方法により、ループ内実行情報を生成するので、説明を省略する。

【 0 0 5 7 】

続いて、ループ処理頻度収集部 1 4 0 は、外側ループ処理のループ内実行情報に基づいて、内側ループ処理頻度、即ち、第 3 ノードが実行される頻度を算出する。例えば、ループ処理頻度収集部 1 4 0 は、外側ループ内構造グラフの第 3 ノードが実行される全ての実行パス、即ち、第 3 から第 8 パスを選択する。そして、ループ処理頻度収集部 1 4 0 は、選択したこれらのパスにおけるループ内実行情報の合計値である 1 9 6 . 1 を、内側ループ処理頻度として生成する。

【 0 0 5 8 】

続いて、ループ内処理頻度収集部 150 は、内側ループ処理頻度、即ち、第 3 ノードが実行される頻度が、予め定められた頻度より高い場合に、以下に示す処理により、内側ループ内処理頻度を収集する。ループ内処理頻度収集部 150 は、内側ループ内構造グラフにおける複数の実行パスのそれぞれが実行される回数を測定するカウンタを起動し、複数の実行パスにおける測定値の合計が予め定められた 100 回となった時点で停止する。そして、ループ内処理頻度収集部 150 は、内側ループ処理が実行される回数に対する、各実行パスが実行される頻度である内側ループ内処理頻度を収集する。

【0059】

例えば、内側ループ処理が実行された回数は、E3 ノードから内側ループに処理が移った回数であるので、第 9 から第 11 実行パスの合計値である 58 回となる。一方、例えば、第 13 実行パスが実行された回数は、40 回である。即ち、ループ内処理頻度収集部 150 は、内側ループ内処理頻度として、内側ループ処理が 58 回実行された場合に、第 13 実行パスが 40 回実行される旨を示す情報を収集する。

【0060】

そして、ループ内実行情報生成部 160 は、内側ループ処理の処理頻度である 196.1 に、内側ループ内処理頻度、例えば、第 13 実行パスの処理頻度である $40/58$ を乗じることにより、内側ループ処理のループ内実行情報である 135.2 を生成する。ループ内実行情報生成部 160 は、第 9 実行パスから第 12 実行パス及び第 14 実行パスにおいても、第 13 実行パスにおける方法と略同一の方法により、ループ内実行情報を生成するので、説明を省略する。

【0061】

以上で説明したループ内実行情報を生成する方法を、数式により示す。

ループ内実行情報生成部 160 は、構造グラフ X における各実行パスの実行情報を、各実行パスが実行された回数である測定値に、以下の式で示す補正係数 C_x を乗じることにより生成する。

【0062】

【数 1】

$$C_X = \begin{cases} thresholdCount(X) & \dots\dots\dots X \text{が外郭グラフの場合} \\ \frac{C_Y \sum_{q \in P_Y(N_X)} C_q}{\sum_{p \in P_X(entry)} C_p} & \dots\dots\dots \text{それ以外} \end{cases}$$

【0063】

ここで、thresholdCount(X)は、構造グラフXに対応付けて予め設定され、構造グラフXにおいて収集する測定値の合計値を示す。また、C_pは、経路pに対して収集された実行回数の測定値、P_x(a)は、構造グラフX中のノードaを通る経路の集合、P_x(entry)は、構造グラフXにおいてループ外からループへ入る経路の集合、N_xは構造グラフXに対応する上位階層の構造グラフ中のループノード、をそれぞれ表す。また、C_yは、構造グラフXの上位階層の構造グラフにおける補正係数である。ここで、上位階層とは、例えば、より外側のループ処理における構造グラフである。例えば、内側ループ内構造グラフの上位階層は、外側ループ内構造グラフであり、外側ループ内構造グラフの上位階層は、外郭構造グラフである。

【0064】

なお、本図は、他の方法により測定され、各実行パスが実際に実行された実行回数を、実行情報に対応付けて更に示す。当該他の方法は、プログラムを10000回実行した場合の、各実行パスが実行された回数を測定している。

【0065】

これに対し、本実施形態に係るコンパイラ装置10は、実行パスが実行される回数を構造グラフ毎に100回、計300回測定することにより、実際の実行回数と略同一な実行情報を生成することができる。従って、コンパイラ装置10は、コンパイル処理に要する時間を短縮することができる。

【0066】

図9(a)は、外郭構造グラフにおいて測定された、各実行パスの実行回数を

示す。プログラムが100回実行された場合に、ノードE1から、第1ノード、第2ノード、及び第7ノードを経由してノードX1に至る第1実行パスは、100回実行される。

【0067】

図9(b)は、外側ループ処理内構造グラフにおいて測定された、各実行パスの実行回数を示す。各実行パスの実行回数の合計が100回となった場合に、ノードE2から、第2ノード、第3ノード、及び第6ノードを経由してノードX2に至る第4実行パスは、50回実行される。また、第6ノード、第2ノード、及び第3ノードを経由して第6ノードに至る第8実行パスは、48回実行される。

【0068】

図9(c)は、内側ループ処理内構造グラフにおいて測定された、各実行パスの実行回数を示す。各実行パスの実行回数の合計が100回となった場合に、ノードE3から、第3ノードを経由してノードX3に至る第11実行パスは、56回実行される。また、第4ノード及び第3ノードを経由して第4ノードに至る第13実行パスは、40回実行される。

【0069】

図9(d)は、ループ内実行情報生成部160により生成されたループ内実行情報の一例を示す。ループ内実行情報生成部160は、プログラムが100回実行された場合における、第8実行パスが実行される頻度を示すループ内実行情報を94.1と生成する。また、ループ内実行情報生成部160は、プログラムが100回実行された場合における、第4実行パスが実行される頻度を示すループ内実行情報を98.0と生成する。また、ループ内実行情報生成部160は、プログラムが100回実行された場合における、第13実行パスが実行される頻度を示すループ内実行情報を135.2と生成する。即ち、コンパイラ装置10によると、コンパイル対象のプログラムは、ループ処理を行わないままプログラムの開始点から終了点まで連続して実行する第4実行パスと、外側ループ処理を繰り返す第8実行パスと、内側ループ処理を繰り返す第13実行パスとを、他の実行パスに比べてより高い頻度で実行する。

【0070】

図10(a)は、最適化部30によりプログラムが最適化される例を示す。最適化部30は、ループ内実行情報に基づき、より高い頻度で実行される複数の実行パス（ホットパス、Hot Path）のそれぞれを、連続した領域に配置するべく、最適化を行う。例えば、最適化部30は、第1ノード、第2ノード、第3ノード、第6ノード、及び第7ノードを連続して実行する第4実行パスを効率的に最適化するべく、外側ループ処理及び内側ループ処理を分離させる。最適化部30は、外側ループ処理として、それぞれに斜線が付された第2ノード、第3ノード、第6ノード、及び第7ノードを、制御フローグラフから分離する。また、最適化部30は、外側ループ処理に対してループピーリング（Loop Peeling）を行うことにより、それぞれに斜線が付された第3ノード及び第6ノードを、内側ループ処理として制御フローグラフから分離する。

【0071】

図10(b)は、最適化部30により最適化されたプログラムにおいて命令列が配置された結果を示す。最適化部30は、第1ノードから第7ノードに至る命令列と、第2ノードから第7ノードに至る命令列と、第4ノードから第6ノードに至る命令列とを、それぞれ連続した領域に配置する。なお、連続して配置されないために実行される分岐処理を矢印で示す。また、各分岐処理における実行情報を矢印に付した。最適化部30は、本図のように、分岐命令により分岐処理が行う頻度を低減することができる。これにより、ハードウェアによる分岐予測処理の効率を高めることができる。更に、プロセッサにおける命令（インストラクション）用のキャッシュメモリのヒット率が向上する。また、冗長な無条件分岐及び冗長な前方分岐の数を減らすことができる。

【0072】

図11は、第1の他の例において、実行情報を生成する方法を示す。本例のコンパイラ装置は、構造グラフを作ることなく、制御フローグラフの全ての実行パスについて、実行頻度を測定する。例えば、本例のコンパイラ装置は、図11の黒点の位置、即ち、第1ノードから第2ノードに至るエッジ、第4ノードから第3ノードに至るエッジ、第4ノードから第5ノードに至るエッジ、第6ノードから第2ノードに至るエッジ、及び第6ノードから第7ノードに至るエッジのそれ

ぞれに、カウンタを挿入し、各カウンタにおける測定値を収集する。

【0073】

図12(a)は、第1の他の例において収集された実行情報の例を制御フローグラフ上に示す。図12(b)は、第1の他の例において収集された実行情報の例を表で示す。本例のコンパイラ装置は、コンパイルに要する時間を短縮するべく、プログラムを300回実行した時点で、カウンタを停止する。本例のコンパイラ装置は、本図に示すとおり、第1ノードから、第2ノード、第3ノード、及び第6ノードを介して第7ノードに至る第3実行パスと、第4ノードから、第3ノード及び第4ノードを介して第3ノードに至る第6実行パスとが、他の実行パスより高い頻度で実行されるホットパスである旨を検出することができる。しかしながら、プログラムを実行させる回数が少ないので、本例のコンパイラ装置は、第6ノードから、第2ノード、第3ノード、及び第6ノードを介して第2ノードに至る第12実行パスが、ホットパスである旨を検出することができない。従って、図10(b)の例とは異なり、第2ノード、第3ノード、第6ノード、及び第7ノードを連続して配置する最適化を行うことができない。

【0074】

これに対し、本実施形態におけるコンパイラ装置10は、第1の他の例と同様コンパイルに要する時間を短縮しつつ、第12実行パスがホットパスであることを検出することができる。

【0075】

図13は、第2の他の例において最適化されたプログラムの例を示す。本例のコンパイラ装置は、制御フローグラフにおける各有向エッジが処理される回数を測定するエッジプロファイル方式により、プログラムの実行情報を収集する。本例によると、第1ノードから、第2ノード、第3ノード、及び第6ノードを介して第7ノードに至る実行パスと、第4ノードから第5ノードに至る実行パスとが、ホットパスであるとして、それぞれ連続した命令列として配置される。しかしながら、第3ノードから第4ノードに至る前方分岐命令及び第4ノードから第3ノードに至る分岐命令が生成され、効率が悪い。

【0076】

これに対し、図10(b)によると、コンパイラ装置10により最適化されたプログラムは、頻繁に実行される前方分岐命令を有しておらず、実行効率が高い。

【0077】

図14は、以上で示した実施形態におけるコンパイラ装置10のハードウェア構成の一例を示す。実施形態又は変形例に係るコンパイラ装置10は、ホストコントローラ1082により相互に接続されるCPU1000、RAM1020、グラフィックコントローラ1075、及び表示装置1080を有するCPU周辺部と、入出力コントローラ1084によりホストコントローラ1082に接続される通信インターフェイス1030、ハードディスクドライブ1040、及びCD-ROMドライブ1060を有する入出力部と、入出力コントローラ1084に接続されるROM1010、フレキシブルディスクドライブ1050、及び入出力チップ1070を有するレガシー入出力部とを備える。

【0078】

ホストコントローラ1082は、RAM1020と、高い転送レートでRAM1020をアクセスするCPU1000及びグラフィックコントローラ1075とを接続する。CPU1000は、ROM1010及びRAM1020に格納されたコンパイラプログラム又は実行時情報生成プログラムに基づいて動作し、各部の制御を行う。グラフィックコントローラ1075は、CPU1000等がRAM1020内に設けたフレームバッファ上に生成する画像データを取得し、表示装置1080上に表示させる。これに代えて、グラフィックコントローラ1075は、CPU1000等が生成する画像データを格納するフレームバッファを、内部に含んでもよい。

【0079】

入出力コントローラ1084は、ホストコントローラ1082と、比較的高速な入出力装置である通信インターフェイス1030、ハードディスクドライブ1040、及びCD-ROMドライブ1060を接続する。通信インターフェイス1030は、ネットワークを介して他の装置と通信する。ハードディスクドライブ1040は、コンパイラ装置10が使用するコンパイラプログラム又は実行時

情報生成プログラム及びデータを格納する。CD-ROMドライブ1060は、CD-ROM1095からコンパイラプログラム若しくは実行時情報生成プログラム又はデータを読み取り、RAM1020を介して入出力チップ1070に提供する。

【0080】

また、入出力コントローラ1084には、ROM1010と、フレキシブルディスクドライブ1050や入出力チップ1070等の比較的低速な入出力装置とが接続される。ROM1010は、コンパイラ装置10の起動時にCPU1000が実行するブートプログラムや、コンパイラ装置10のハードウェアに依存するプログラム等を格納する。フレキシブルディスクドライブ1050は、フレキシブルディスク1090からコンパイラプログラム若しくは実行時情報生成プログラム又はデータを読み取り、RAM1020を介して入出力チップ1070に提供する。入出力チップ1070は、フレキシブルディスク1090や、例えばパラレルポート、シリアルポート、キーボードポート、マウスポート等を介して各種の入出力装置を接続する。

【0081】

コンパイラ装置10に提供されるコンパイラプログラム又は実行時情報生成プログラムは、フレキシブルディスク1090、CD-ROM1095、又はICカード等の記録媒体に格納されて利用者によって提供される。コンパイラプログラム又は実行時情報生成プログラムは、記録媒体から読み出され、入出力チップ1070を介してコンパイラ装置10にインストールされ、コンパイラ装置10において実行される。

【0082】

コンパイラ装置10にインストールされて実行されるコンパイラプログラム又は実行時情報生成プログラムは、制御フローグラフ生成モジュールと、ループ検出モジュールと、構造グラフ生成モジュールと、カウンタ挿入モジュールと、ループ処理頻度収集モジュールと、ループ内処理頻度収集モジュールと、ループ内実行情報生成モジュールと、最適化モジュールとを含む。各モジュールがコンパイラ装置10に働きかけて行わせる動作は、図1から図13において説明したコ

ンパイラ装置10における、対応する部材の動作と同一であるから、説明を省略する。

【0083】

以上に示したプログラム又はモジュールは、外部の記憶媒体に格納されてもよい。記憶媒体としては、フレキシブルディスク1090、CD-ROM1095の他に、DVDやPD等の光学記録媒体、MD等の光磁気記録媒体、テープ媒体、ICカード等の半導体メモリ等を用いることができる。また、専用通信ネットワークやインターネットに接続されたサーバシステムに設けたハードディスク又はRAM等の記憶装置を記録媒体として使用し、ネットワークを介してコンパイラプログラム又は実行時情報生成プログラムをコンパイラ装置10に提供してもよい。

【0084】

上記説明から明らかなように、コンパイラ装置10は、ループ内実行情報を高速に収集し、プログラムを適切に最適化することができる。例えば、コンパイラ装置10が実行時コンパイラである場合には、コンパイルに多くの時間をかけることができないので、より効率的にプログラムを最適化することができる。

【0085】

以上、本発明を実施形態を用いて説明したが、本発明の技術的範囲は上記実施形態に記載の範囲には限定されない。上記実施形態に、多様な変更または改良を加えることができる。そのような変更または改良を加えた形態も本発明の技術的範囲に含まれ得ることが、特許請求の範囲の記載から明らかである。

【0086】

以上に示した実施形態によると、各項目に示すコンパイラ装置、コンパイラプログラム、記録媒体、コンパイル方法、実行時情報生成装置、及び実行時情報生成プログラムが実現される。

【0087】

(項目1) 最適化対象のプログラムにおける各処理が実行される頻度を収集し、収集した当該頻度に基づいて前記プログラムを最適化するコンパイラ装置であって、前記プログラムのうち、繰り返し実行されるループ処理を検出するループ

処理検出部と、前記プログラムにおいて、前記ループ処理が実行されるループ処理頻度を収集するループ処理頻度収集部と、前記ループ処理が実行される回数に対する、前記ループ処理に含まれる複数のループ内処理のそれぞれが実行される頻度であるループ内処理頻度を収集するループ内処理頻度収集部と、前記ループ処理頻度及び前記ループ内処理頻度に基づき、前記プログラムが実行された場合における、前記複数のループ内処理のそれぞれが実行される頻度を示すループ内実行情報を生成するループ内実行情報生成部と、前記ループ内実行情報生成部により生成された前記ループ内実行情報に基づき、前記プログラムを最適化する最適化部とを備えるコンパイラ装置。

（項目 2） 前記ループ内処理頻度収集部は、前記ループ処理頻度が予め定められた頻度より高い場合に、前記ループ内処理頻度を収集する項目 1 記載のコンパイラ装置。

【 0 0 8 8 】

（項目 3） 前記ループ内実行情報生成部は、前記ループ処理頻度に前記ループ内処理頻度を乗じることにより、前記ループ内実行情報を生成する項目 1 記載のコンパイラ装置。

（項目 4） 前記ループ処理は、更に内側のループ処理である内側ループ処理を含む外側ループ処理であり、前記ループ処理検出部は、前記内側ループ処理を更に検出し、前記ループ処理頻度収集部は、前記プログラムにおいて、前記内側ループ処理が実行されるループ処理頻度を前記ループ内実行情報に基づいて更に収集し、前記ループ内処理頻度収集部は、前記内側ループ処理のループ内処理頻度を収集し、前記ループ内実行情報生成部は、前記内側ループ処理におけるループ内処理頻度に、前記内側ループ処理のループ処理頻度を乗じることにより、前記内側ループ処理のループ内実行情報を生成する項目 1 記載のコンパイラ装置。

（項目 5） 前記ループ処理頻度収集部は、前記ループ処理が実行される回数を測定するカウンタを、前記プログラムが予め定められた回数実行された時点で停止して、当該カウンタにより測定された回数を前記ループ処理頻度として収集し、前記ループ内処理頻度収集部は、前記複数のループ内処理のそれぞれが実行される回数を測定するカウンタを、前記複数のループ内処理の測定値の合計が予め

定められた回数となった時点で停止する項目 1 記載のコンパイラ装置。

【 0 0 8 9 】

(項目 6) 前記プログラムにおける、複数の命令列のそれぞれをノードとして生成し、前記複数の命令列間の実行順序を前記ノード間の有向エッジとして生成した制御フローグラフを生成する制御フローグラフ生成部と、前記制御フローグラフにおいて、前記ループ処理を形成するノードの集合に代えて、前記ループ処理全体を示す単一のループノードを生成した外郭構造グラフと、前記ループ処理を形成するノードの集合の制御フローグラフであるループ内構造グラフとを生成する構造グラフ生成部と、前記外郭構造グラフ及び前記ループ内構造グラフのそれぞれにおいて、当該構造グラフにおける各実行パスが実行される回数をカウントするべく、前記プログラムにカウンタを挿入するカウンタ挿入部とを更に備え、前記ループ処理頻度収集部は、前記プログラムが実行される回数に対する、前記ループノードが実行される回数を、前記ループ処理頻度として生成し、前記ループ内処理頻度収集部は、前記ループ処理が実行される回数に対する、前記ループ内構造グラフにおける各実行パスが実行された回数を、前記ループ内処理頻度として収集する項目 1 記載のコンパイラ装置。

(項目 7) 前記ループ処理頻度収集部は、前記プログラムが予め定められた回数実行された場合において、前記ループノードを含む実行パスが実行される回数をカウントするべく挿入されたカウンタの測定値を、ループ処理頻度として収集し、前記ループ内処理頻度収集部は、前記複数のループ内処理の合計測定値が予め定められた回数となった場合において、前記ループ内構造グラフにおける各実行パスが実行される回数をカウントするべく挿入されたカウンタの測定値に基づき、ループ内処理頻度を収集する項目 6 記載のコンパイラ装置。

【 0 0 9 0 】

(項目 8) 前記カウンタ挿入部は、前記外郭構造グラフの各実行パスが実行される回数を測定するべくカウンタを挿入する前記プログラム中の挿入位置が、前記ループ内構造グラフの各実行パスが実行される回数を測定するべくカウンタを挿入する前記プログラム中の位置と同一であり、かつ前記外郭構造グラフ及び前記ループ内構造グラフのうち多くとも一方におけるカウンタが起動される場合に

、当該挿入位置において、前記外郭構造グラフ及び前記ループ内構造グラフの双方における実行パスが実行される回数を測定するカウンタを挿入する項目 6 記載のコンパイラ装置。

(項目 9) 前記カウンタ挿入部は、前記外郭構造グラフの各実行パスが実行される回数を測定するべくカウンタを挿入する前記プログラム中の挿入位置が、前記ループ内構造グラフの各実行パスが実行される回数を測定するべくカウンタを挿入する前記プログラム中の位置と同一であり、かつ前記外郭構造グラフ及び前記ループ内構造グラフのうち多くとも一方におけるカウンタが起動される場合に、前記外郭構造グラフ及び前記ループ内構造グラフのそれぞれにおいて各実行パスの実行の回数を測定する複数の測定処理を生成し、前記ループ内処理頻度収集部は、処理を他の部分に移すジャンプ命令を前記挿入位置に挿入し、当該ジャンプ命令のジャンプ先を、前記複数の測定処理の何れかに設定することにより、前記外郭構造グラフ及び前記ループ内構造グラフの双方における実行パスが実行される回数を測定する項目 6 記載のコンパイラ装置。

【 0 0 9 1 】

(項目 1 0) 前記ループ処理は、更に内側のループ処理である内側ループ処理を含む外側ループ処理であり、前記ループ処理検出部は、前記内側ループ処理を更に検出し、前記構造グラフ生成部は、前記外側ループ処理の制御フローグラフにおいて、前記内側ループ処理を形成するノードの集合に代えて、単一の内側ループノードを生成したグラフを外側ループ内構造グラフとして生成し、前記内側ループ処理を構成するノードの集合の制御フローグラフである内側ループ内構造グラフを生成し、前記カウンタ挿入部は、前記内側ループ内構造グラフの各実行パスが実行される回数を測定するカウンタを更に挿入し、前記ループ処理頻度収集部は、前記プログラムにおいて、前記内側ループ処理が実行されるループ処理頻度を前記ループ内実行情報に基づいて更に収集し、前記ループ内処理頻度収集部は、前記内側ループ処理が実行される回数に対する、前記内側ループ内構造グラフにおける各実行パスが実行される頻度を、前記内側ループ処理におけるループ内処理頻度として収集し、前記ループ内実行情報生成部は、前記内側ループ処理におけるループ内実行情報を、前記内側ループ処理におけるループ内処理頻度

に、前記内側ループ処理のループ処理頻度を乗じることにより更に生成する項目 6 記載のコンパイラ装置。

(項目 1 1) 前記カウンタ挿入部は、前記外側ループ内構造グラフの各実行パスが実行される回数を測定するべくカウンタを挿入する前記プログラム中の挿入位置が、前記内側ループ内構造グラフの各実行パスが実行される回数を測定するべくカウンタを挿入する前記プログラム中の位置と同一であり、かつ前記外側ループ内構造グラフ及び前記内側ループ内構造グラフのうち多くとも一方におけるカウンタが起動される場合に、当該挿入位置において、前記外側ループ内構造グラフ及び前記内側ループ内構造グラフの双方における実行パスが実行される回数を測定するカウンタを挿入する項目 1 0 記載のコンパイラ装置。

(項目 1 2) 前記カウンタ挿入部は、前記外側ループ内構造グラフの各実行パスが実行される回数を測定するべくカウンタを挿入する前記プログラム中の挿入位置が、前記内側ループ内構造グラフの各実行パスが実行される回数を測定するべくカウンタを挿入する前記プログラム中の位置と同一であり、かつ前記外側ループ内構造グラフ及び前記内側ループ内構造グラフのうち多くとも一方におけるカウンタが起動される場合に、前記外側ループ内構造グラフ及び前記内側ループ内構造グラフのそれぞれにおいて各実行パスの実行の回数を測定する複数の測定処理を生成し、前記ループ内処理頻度収集部は、処理を他の部分に移すジャンプ命令を前記挿入位置に挿入し、当該ジャンプ命令のジャンプ先を、前記複数の測定処理の何れかに設定することにより、前記外側ループ内構造グラフ及び前記内側ループ内構造グラフの双方における実行パスが実行される回数を測定する項目 1 0 記載のコンパイラ装置。

(項目 1 3) 最適化対象のプログラムにおける各処理が実行される頻度を収集し、収集した当該頻度に基づいて前記プログラムを最適化するコンパイラ装置としてコンピュータを機能させるコンパイラプログラムであって、前記コンピュータを、前記プログラムのうち、繰り返し実行されるループ処理を検出するループ処理検出部と、前記プログラムにおいて、前記ループ処理が実行されるループ処理頻度を収集するループ処理頻度収集部と、前記ループ処理が実行される回数に対する、前記ループ処理に含まれる複数のループ内処理のそれぞれが実行される

頻度であるループ内処理頻度を収集するループ内処理頻度収集部と、前記ループ処理頻度及び前記ループ内処理頻度に基づき、前記プログラムが実行された場合における、前記複数のループ内処理のそれぞれが実行される頻度を示すループ内実行情報を生成するループ内実行情報生成部と、前記ループ内実行情報生成部により生成された前記ループ内実行情報に基づき、前記プログラムを最適化する最適化部として機能させるコンパイラプログラム。

【 0 0 9 2 】

(項目 1 4) 項目 1 3 に記載のコンパイラプログラムを記録した記録媒体。

(項目 1 5) 最適化対象のプログラムにおける各処理が実行される頻度を収集し、収集した当該頻度に基づいて前記プログラムを最適化するコンパイル方法であって、前記プログラムのうち、繰り返し実行されるループ処理を検出するループ処理検出段階と、前記プログラムにおいて、前記ループ処理が実行されるループ処理頻度を収集するループ処理頻度収集段階と、前記ループ処理が実行される回数に対する、前記ループ処理に含まれる複数のループ内処理のそれぞれが実行される頻度であるループ内処理頻度を収集するループ内処理頻度収集段階と、前記ループ処理頻度及び前記ループ内処理頻度に基づき、前記プログラムが実行された場合における、前記複数のループ内処理のそれぞれが実行される頻度を示すループ内実行情報を生成するループ内実行情報生成段階と、前記ループ内実行情報生成部により生成された前記ループ内実行情報に基づき、前記プログラムを最適化する最適化段階とを備えるコンパイル方法。

【 0 0 9 3 】

(項目 1 6) 最適化対象のプログラムにおける各処理が実行される頻度を収集する実行時情報生成装置であって、前記プログラムのうち、繰り返し実行されるループ処理を検出するループ処理検出部と、前記プログラムにおいて、前記ループ処理が実行されるループ処理頻度を収集するループ処理頻度収集部と、前記ループ処理が実行される回数に対する、前記ループ処理に含まれる複数のループ内処理のそれぞれが実行される頻度であるループ内処理頻度を収集するループ内処理頻度収集部と、前記ループ処理頻度及び前記ループ内処理頻度に基づき、前記プログラムが実行された場合における、前記複数のループ内処理のそれぞれが実

行される頻度を示すループ内実行情報を生成するループ内実行情報生成部とを備え、前記ループ内実行情報生成部により生成された前記ループ内実行情報に基づき、前記プログラムを最適化させる実行時情報生成装置。

(項目 1 7) 最適化対象のプログラムにおける各処理が実行される頻度を収集する実行時情報生成装置としてコンピュータを機能させる実行時情報生成プログラムであって、前記コンピュータを、前記プログラムのうち、繰り返し実行されるループ処理を検出するループ処理検出部と、前記プログラムにおいて、前記ループ処理が実行されるループ処理頻度を収集するループ処理頻度収集部と、前記ループ処理が実行される回数に対する、前記ループ処理に含まれる複数のループ内処理のそれぞれが実行される頻度であるループ内処理頻度を収集するループ内処理頻度収集部と、前記ループ処理頻度及び前記ループ内処理頻度に基づき、前記プログラムが実行された場合における、前記複数のループ内処理のそれぞれが実行される頻度を示すループ内実行情報を生成するループ内実行情報生成部として機能させ、前記ループ内実行情報生成部により生成された前記ループ内実行情報に基づき、前記プログラムを最適化させる実行時情報生成プログラム。

(項目 1 8) 項目 1 7 に記載の実行時情報生成プログラムを記録した記録媒体

【 0 0 9 4 】

【発明の効果】

上記説明から明らかなように、本発明によれば、プログラムの各処理が実行される頻度を収集する処理を、高速に行うことができる。

【図面の簡単な説明】

【図 1】

図 1 は、コンパイラ装置 1 0 の機能ブロック図を示す。

【図 2】

図 2 は、コンパイラ装置 1 0 のフローチャートを示す。

【図 3】

図 3 は、最適化対象のプログラムの一例を示す。

【図 4】

図 4 は、制御フローグラフの一例を示す。

【図 5】

図 5 (a) は、構造グラフを生成する対象となる制御フローグラフの一例を示す。

図 5 (b) は、制御フローグラフの実行パスを示す。

図 5 (c) は、制御フローグラフから生成した構造グラフの実行パスを示す。

【図 6】

図 6 (a) は、図 4 に示した制御フローグラフから生成される外郭構造グラフの一例を示す。

図 6 (b) は、図 4 に示した制御フローグラフから生成される外側ループ内構造グラフの一例を示す。

図 6 (c) は、図 4 に示した制御フローグラフから生成される内側ループ内構造グラフの一例を示す。

【図 7】

図 7 (a) は、プログラムに挿入されたカウンタが停止されている例を示す。

図 7 (b) は、プログラムに挿入されたカウンタが起動されている例を示す。

図 7 (c) は、同一の挿入位置に複数のカウンタを生成する例を示す。

【図 8】

図 8 は、コンパイラ装置 1 0 が生成した実行情報の一例を示す。

【図 9】

図 9 (a) は、外郭構造グラフにおいて測定された、各実行パスの実行回数を示す。

図 9 (b) は、外側ループ処理内構造グラフにおいて測定された、各実行パスの実行回数を示す。

図 9 (c) は、内側ループ処理内構造グラフにおいて測定された、各実行パスの実行回数を示す。

図 9 (d) は、ループ内実行情報生成部 1 6 0 により生成されたループ内実行情報の一例を示す。

【図 1 0】

図 1 0 (a) は、最適化部 3 0 によりプログラムが最適化される例を示す。

図 1 0 (b) は、最適化部 3 0 により最適化されたプログラムにおいて命令列が配置された結果を示す。

【図 1 1】

図 1 1 は、第 1 の他の例における実行情報の一例を示す。

【図 1 2】

図 1 2 (a) は、第 1 の他の例において収集された実行情報の例を制御フローグラフ上に示す。

図 1 2 (b) は、第 1 の他の例において収集された実行情報の例を表で示す。

【図 1 3】

図 1 3 は、第 2 の他の例において最適化されたプログラムの例を示す。

【図 1 4】

図 1 4 は、以上で示した実施形態におけるコンパイラ装置 1 0 のハードウェア構成の一例を示す。

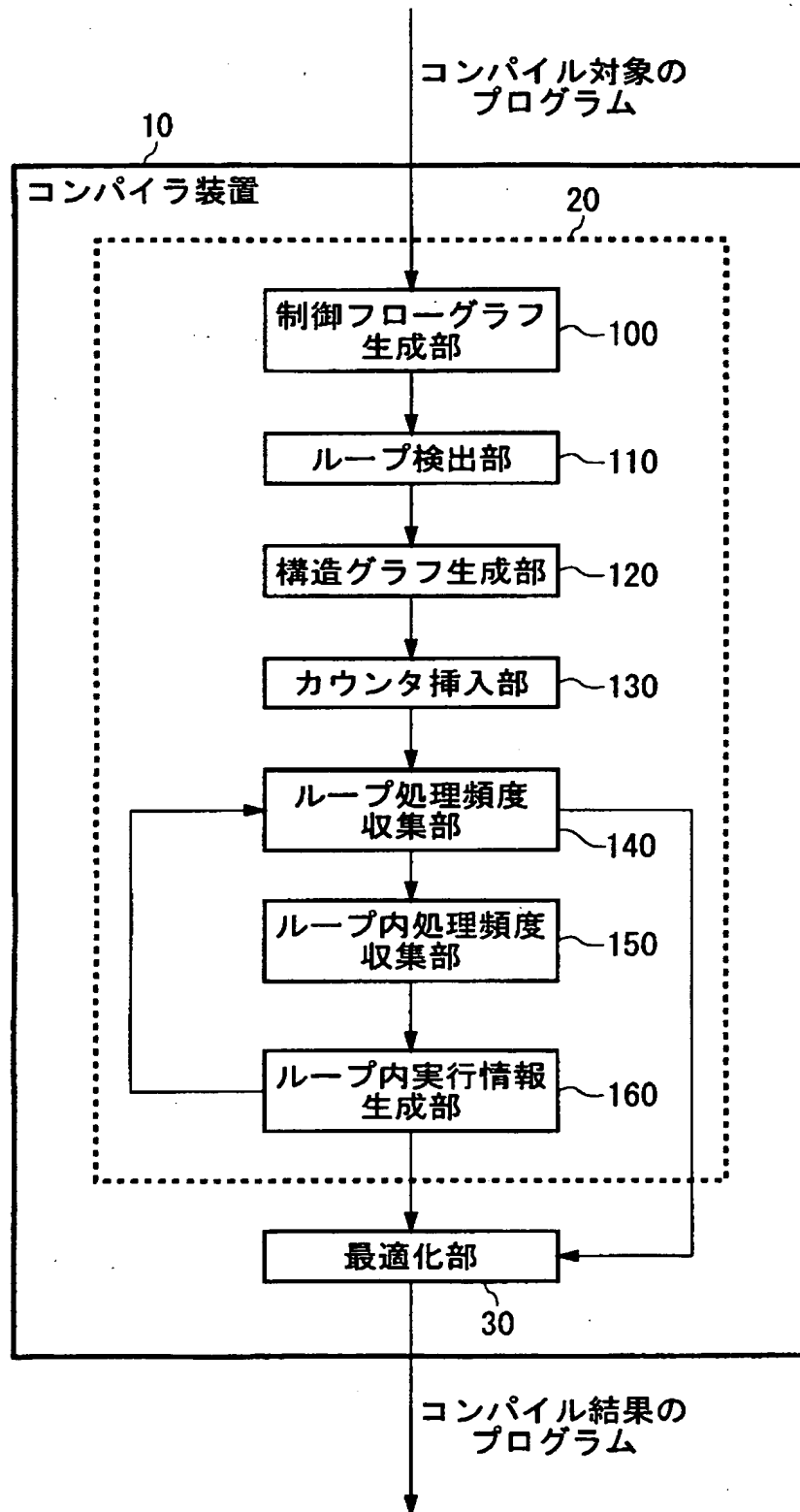
【符号の説明】

- 1 0 コンパイラ装置
- 2 0 実行時情報生成装置
- 3 0 最適化部
- 1 0 0 制御フローグラフ生成部
- 1 1 0 ループ検出部
- 1 2 0 構造グラフ生成部
- 1 3 0 カウンタ挿入部
- 1 4 0 ループ処理頻度収集部
- 1 5 0 ループ内処理頻度収集部
- 1 6 0 ループ内実行情報生成部
- 5 0 0 ヘッダノード
- 5 1 0 ラッチノード
- 5 2 0 実行パス
- 5 3 0 実行パス

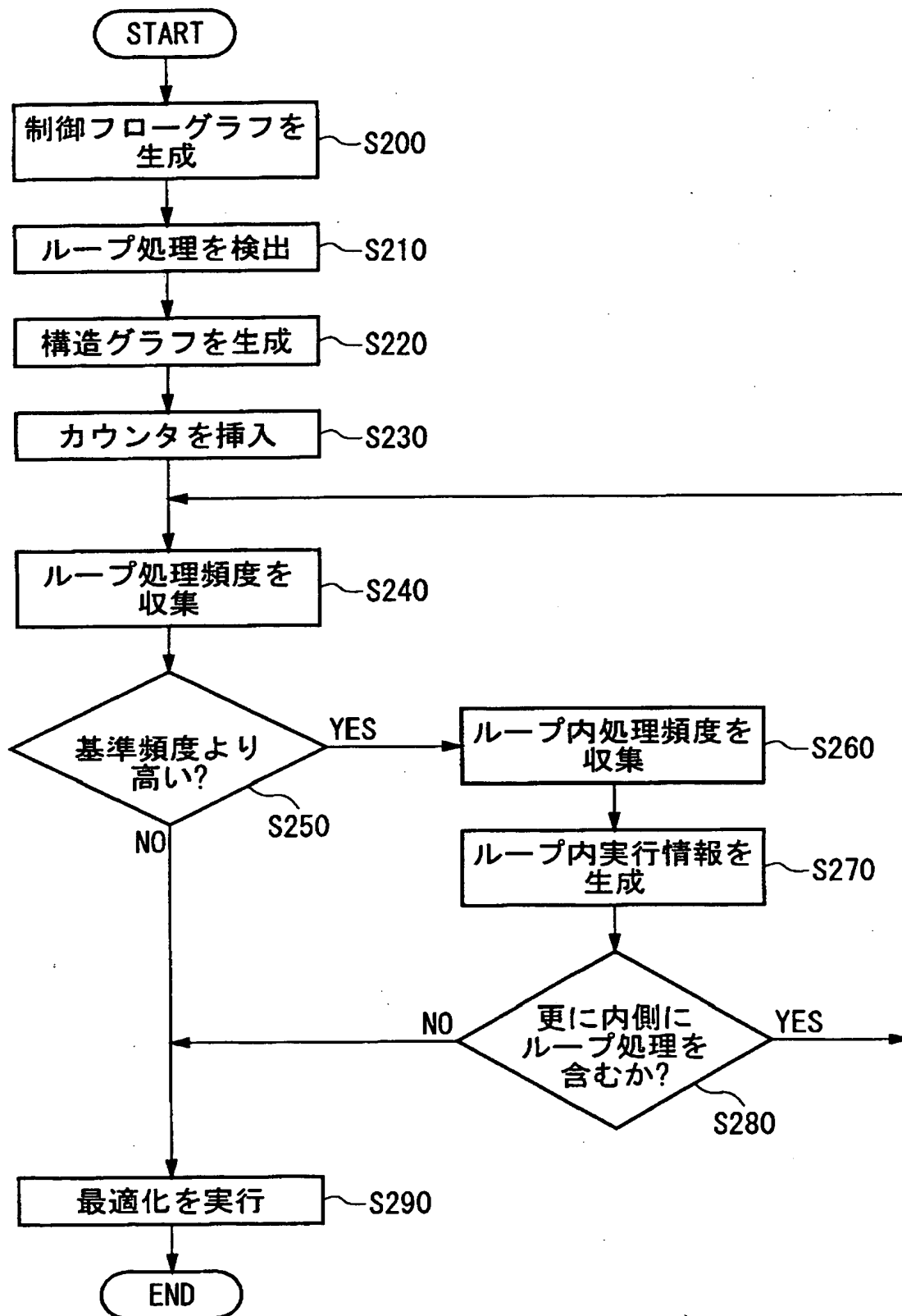
5 4 0 実行パス
5 5 0 実行パス
5 6 0 実行パス
7 0 0 NOP 命令
7 1 0 測定処理
7 2 0 ジャンプ命令
7 3 0 測定処理

【書類名】 図面

【図 1】



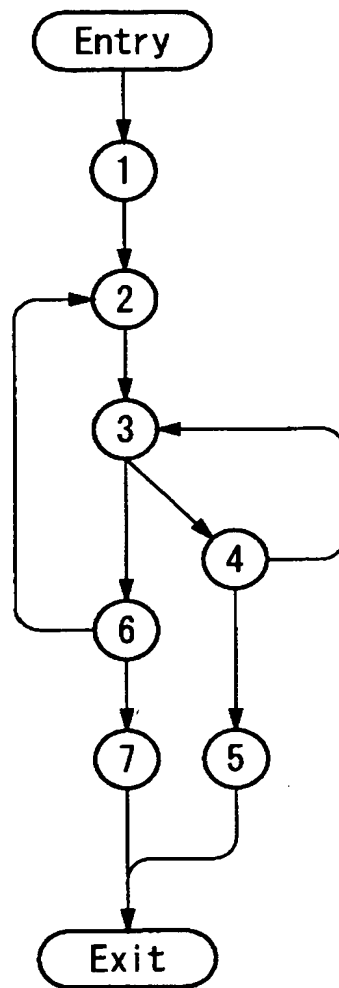
【図 2】



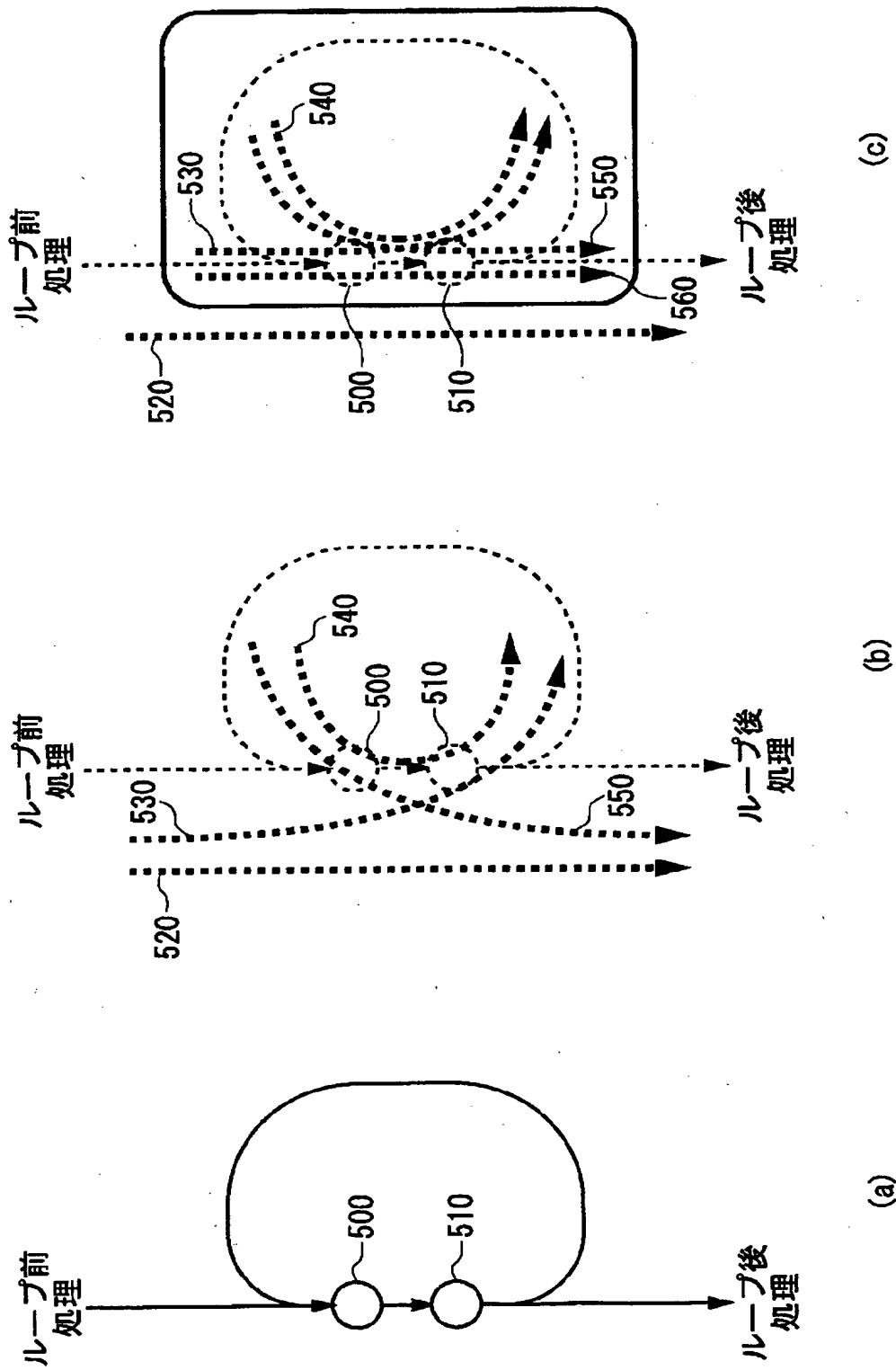
【図 3】

```
1: public void m()
2: {
3:     ... ..... 第1基本ブロック
4:     do {
5:         ... ..... 第2基本ブロック
6:         while (...) { ..... 第3基本ブロック
7:             ... } ..... 第4基本ブロック
8:             if (...) } ..... 第4基本ブロック
9:             return; ..... 第5基本ブロック
10:    }
11:    ... } ..... 第6基本ブロック
12:    } while (...) } ..... 第6基本ブロック
13:    ... ..... 第7基本ブロック
14: }
```

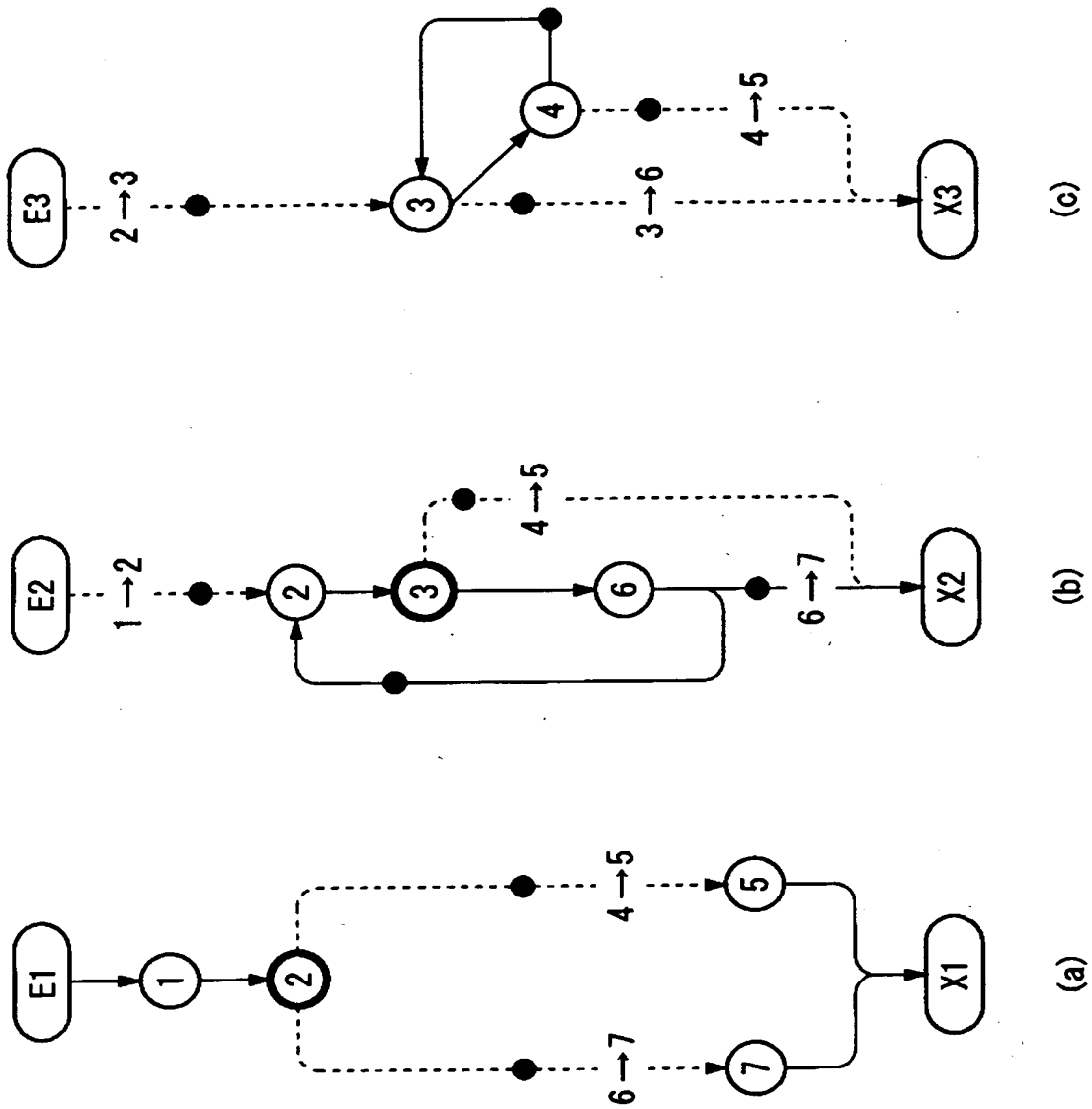
【図 4】



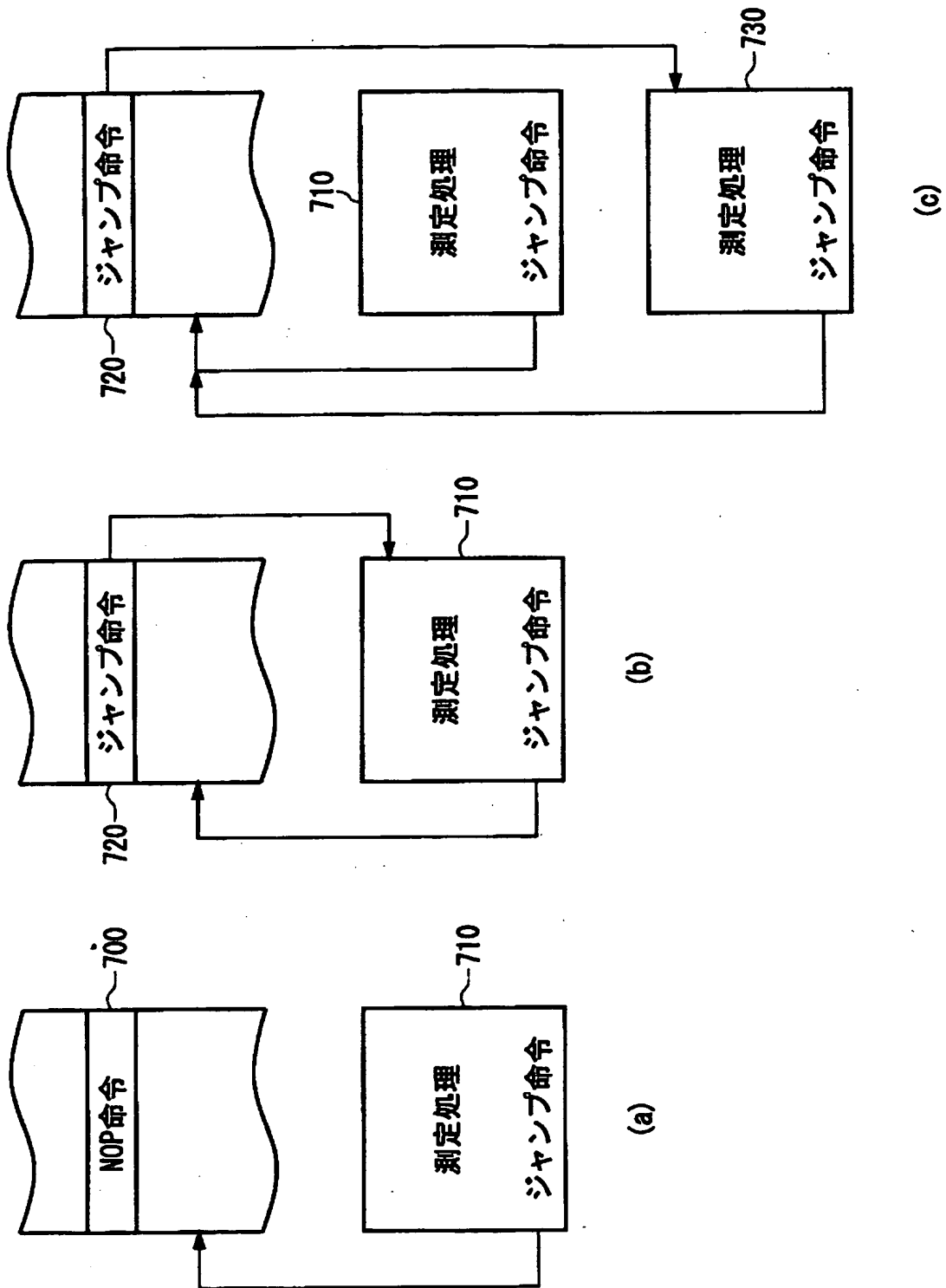
【図5】



【図 6】



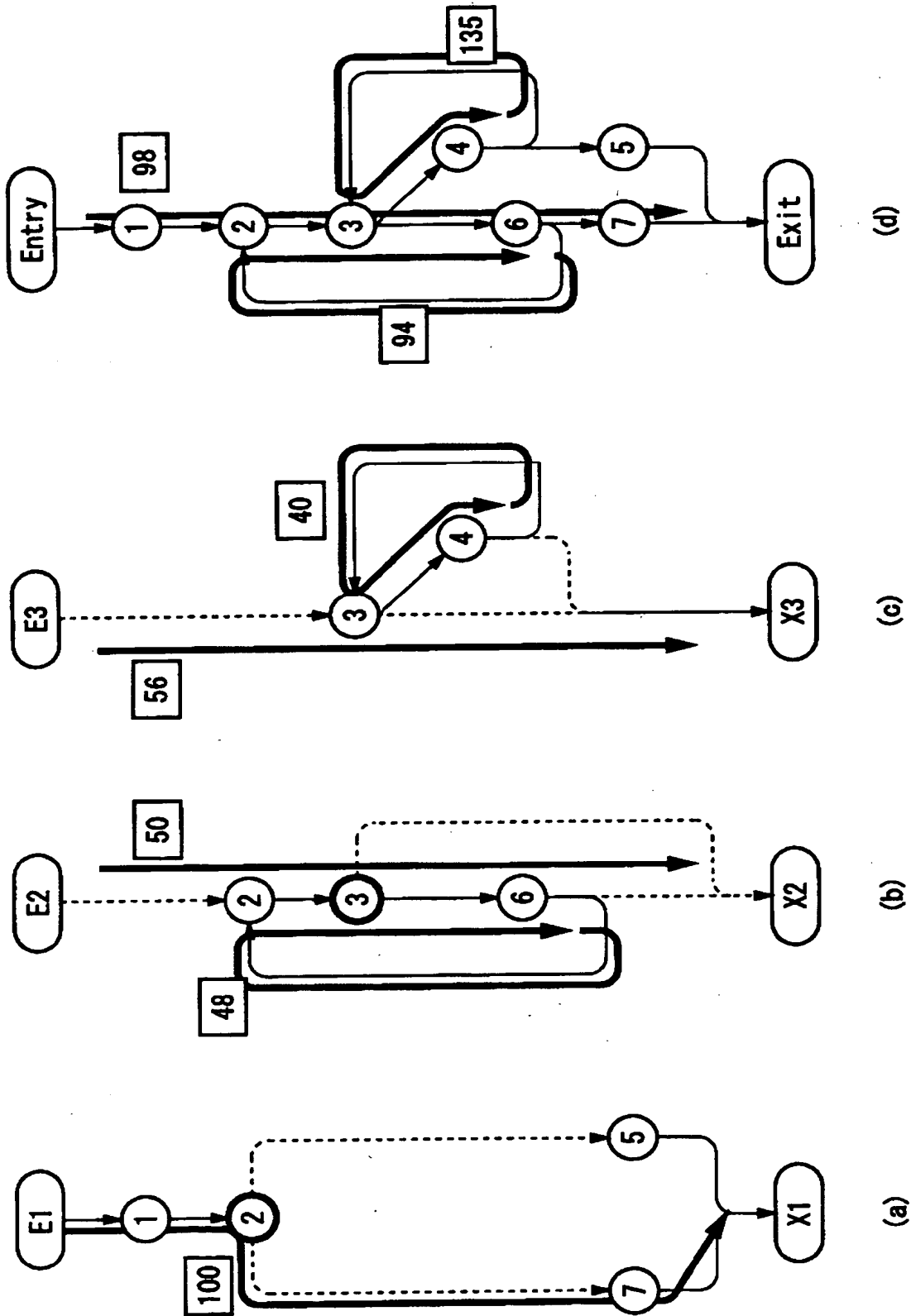
【図 7】



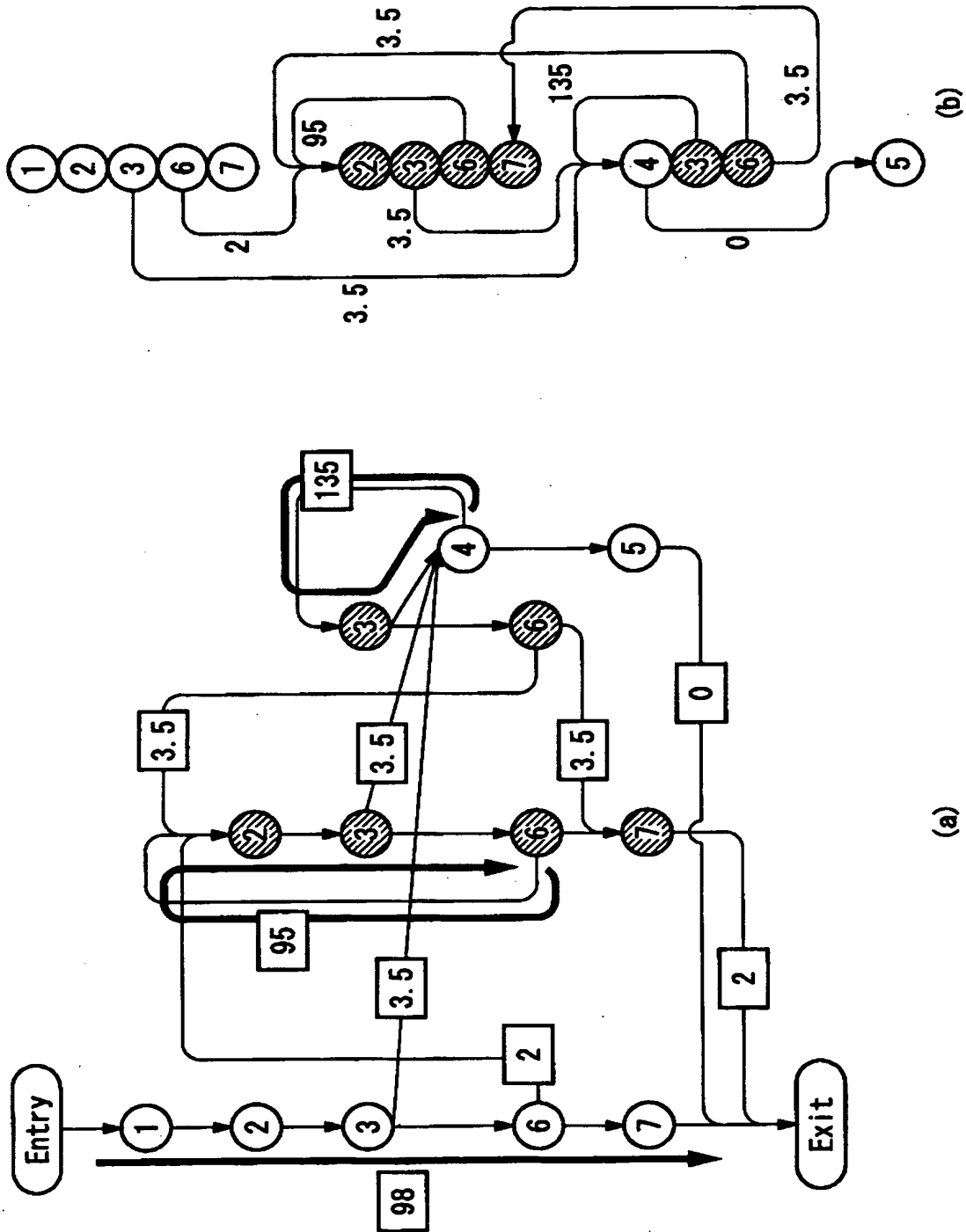
【図 8】

構造グラフ	構造グラフにおける実行パス		測定値	実行情報	実際の 実行回数
	識別 情報	実行パス			
外郭構造グラフ	1	E1-1-2-7-X1	100	100.0	10000
	2	E1-1-2-5-X1	0	0.0	0
外側ループ内 構造グラフ	3	E2-2-3-X2	0	0.0	0
	4	E2-2-3-6-X2	50	98.0	9800
	5	E2-2-3-6-(2)	1	2.0	200
	6	(6)-2-3-X2	0	0.0	0
	7	(6)-2-3-6-X2	1	2.0	200
	8	(6)-2-3-6-(2)	48	94.1	9600
内側ループ内 構造グラフ	9	E3-3-4-X3	0	0.0	0
	10	E3-3-4-(3)	2	6.8	683
	11	E3-3-X3	56	189.3	19117
	12	(4)-3-4-X3	0	0.0	0
	13	(4)-3-4-(3)	40	135.2	13660
	14	(4)-3-X3	2	6.8	683
合計			300	634.1	63943

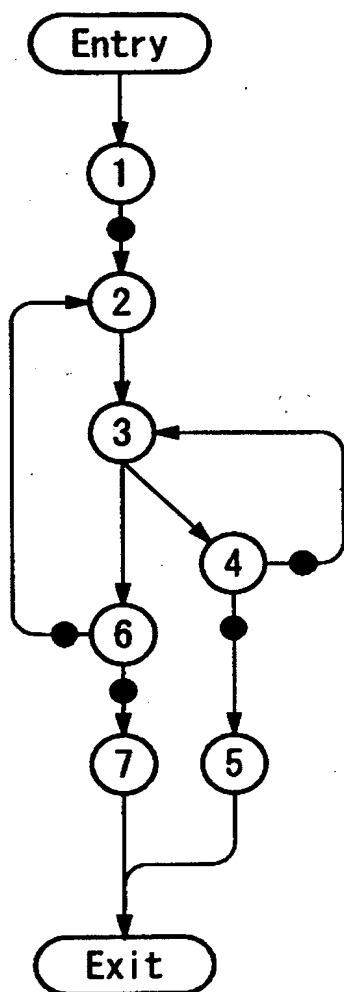
【図 9】



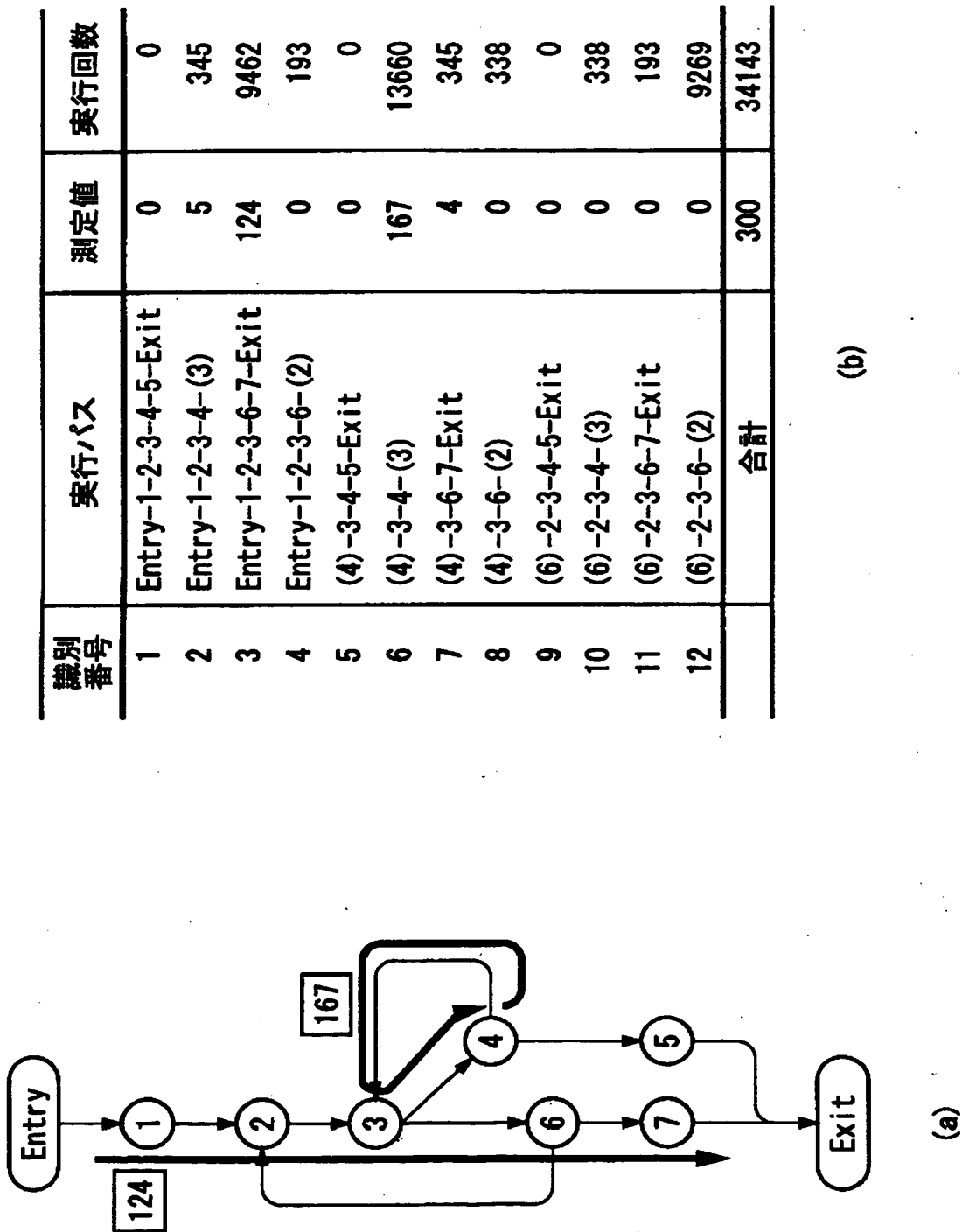
【図 10】



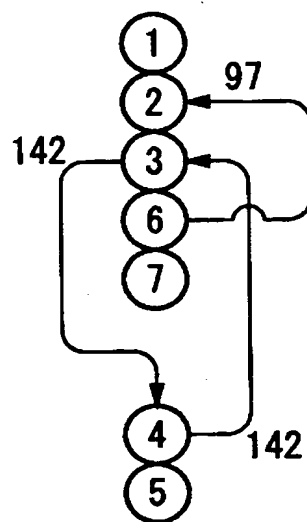
【図 11】



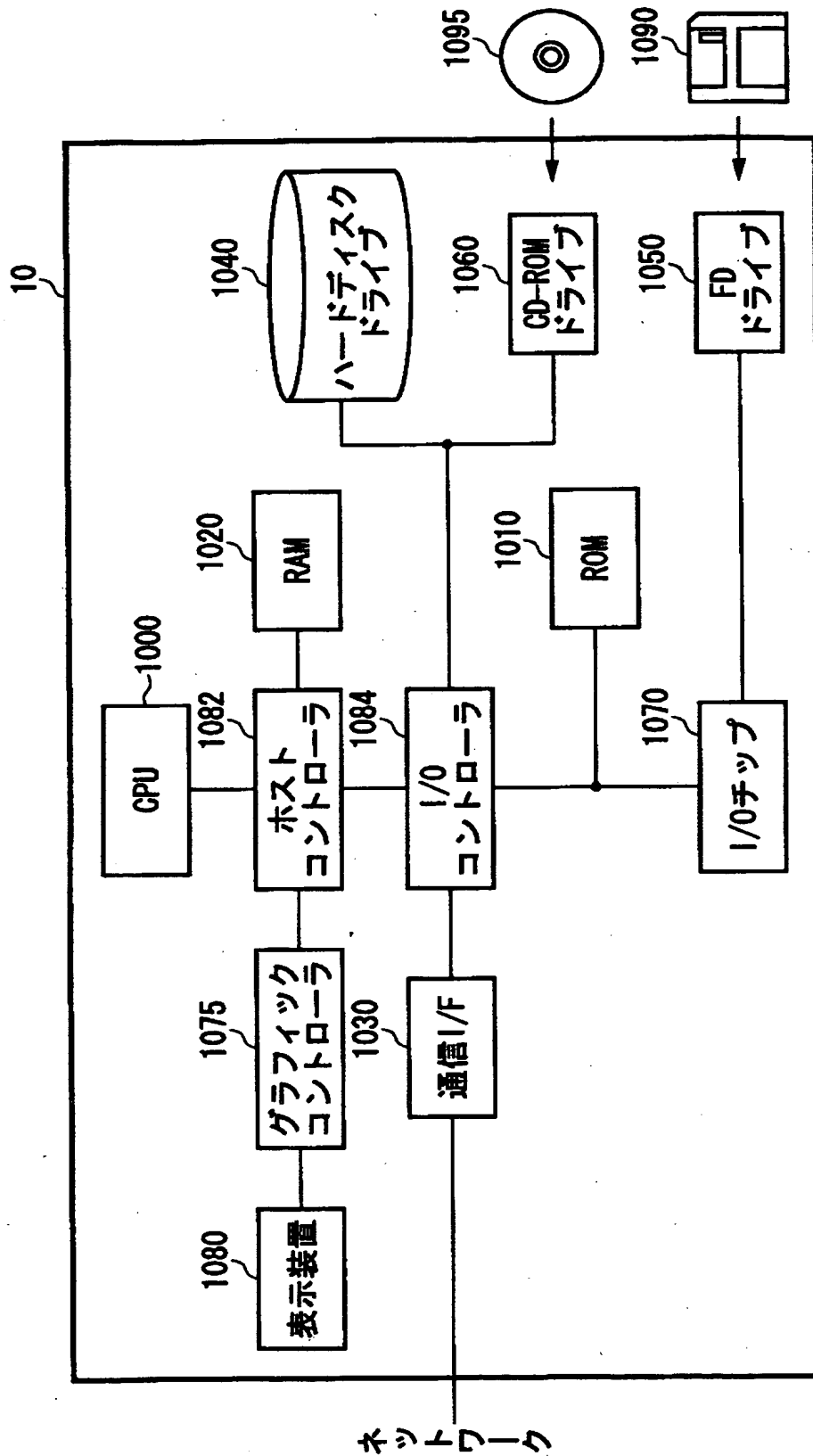
【図 12】



【図 1 3】



【図14】



【書類名】 要約書

【要約】

【課題】 プログラムの各処理が実行される頻度を高速に収集する。

【解決手段】 最適化対象のプログラムにおける各処理が実行される頻度に基づいてプログラムを最適化するコンパイラ装置は、プログラムのうち、繰り返し実行されるループ処理を検出するループ処理検出部と、プログラムにおいて、ループ処理が実行されるループ処理頻度を収集するループ処理頻度収集部と、ループ処理が実行される回数に対する、ループ処理に含まれる複数のループ内処理のそれぞれが実行される頻度であるループ内処理頻度を収集するループ内処理頻度収集部と、ループ処理頻度及びループ内処理頻度に基づき、プログラムが実行された場合における、複数のループ内処理のそれぞれが実行される頻度を示すループ内実行情報を生成するループ内実行情報生成部と、ループ内実行情報に基づき、プログラムを最適化する最適化部とを備える。

【選択図】 図 1

認定・付加情報

特許出願の番号	特願 2003-066010
受付番号	50300398960
書類名	特許願
担当官	伊藤 雅美 2132
作成日	平成15年 4月24日

<認定情報・付加情報>

【特許出願人】

【識別番号】	390009531
【住所又は居所】	アメリカ合衆国10504、ニューヨーク州 アーモンク ニュー オーチャード ロード
【氏名又は名称】	インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】

【識別番号】	100086243
【住所又は居所】	神奈川県大和市下鶴間1623番地14 日本アイ・ビー・エム株式会社 大和事業所内
【氏名又は名称】	坂口 博

【代理人】

【識別番号】	100091568
【住所又は居所】	神奈川県大和市下鶴間1623番地14 日本アイ・ビー・エム株式会社 大和事業所内
【氏名又は名称】	市位 嘉宏

【代理人】

【識別番号】	100108501
【住所又は居所】	神奈川県大和市下鶴間1623番14 日本アイ・ビー・エム株式会社 知的所有権
【氏名又は名称】	上野 剛史

【復代理人】

申請人	
【識別番号】	100104156
【住所又は居所】	東京都新宿区新宿1丁目24番12号 東信ビル 6階 龍華国際特許事務所
【氏名又は名称】	龍華 明裕

次頁無

出 願 人 履 歴 情 報

識別番号 [390009531]

1. 変更年月日 2002年 6月 3日

[変更理由] 住所変更

住 所 アメリカ合衆国10504、ニューヨーク州 アーモンク ニ
ュー オーチャード ロード

氏 名 インターナショナル・ビジネス・マシーンズ・コーポレーショ
ン